

# Trabajo Fin de Grado

Estudio de Viabilidad del Desarrollo de una BCI  
Aplicada a un Sistema Informático

BCI Development Feasibility Research Applied to  
a Computer System

Autor

Iñigo Garaboa Cotelo

Director

Dr. David Asiain Ansorena

Escuela Universitaria Politécnica La Almunia  
2020





**Escuela Universitaria  
Politécnica - La Almunia**  
Centro adscrito  
**Universidad Zaragoza**

**ESCUELA UNIVERSITARIA POLITÉCNICA  
DE LA ALMUNIA DE DOÑA GODINA (ZARAGOZA)**

<b>MEMORIA</b>
----------------

Estudio de Viabilidad del Desarrollo de una  
BCI Aplicada a un Sistema Informático

BCI Development Feasibility Research  
Applied to a Computer System

424.20.48

Autor: Iñigo Garaboa Cotelo

Director: Dr. David Asiain Ansorena

Fecha: 25 de noviembre de 2020



## AGRADECIMIENTOS

En primer lugar, quiero agradecer a mi tutor, el Dr. David Asiain Ansorena, por haberse ofrecido a dirigir mi Trabajo de Fin de Grado, aportándome consejo y ayuda tanto en la planificación del proyecto, como en la labor de adquisición de datos.

Por otro lado, me gustaría dar las gracias al Instituto IBERMÁTICA de Innovación I3B por haberme dado la oportunidad de realizar este proyecto y haberme provisto del equipo necesario. Más concretamente, quiero expresar mi especial agradecimiento al Dr. Aitor Moreno Fernández de Leceta, director del área de Inteligencia Artificial, ya que sin su implicación hubiera sido muy complicado lograr el resultado obtenido.

Finalmente, agradecer a mi familia y amigos por el apoyo recibido a lo largo de estos últimos cuatro años a través de los cuales no solo he crecido intelectualmente, sino también como persona.



## INDICE BREVE

1. RESUMEN	1
2. ABSTRACT	2
3. INTRODUCCIÓN	3
4. FUNDAMENTOS TEÓRICOS Y ESTADO DE ARTE	5
5. MÉTODO OPERATIVO	46
6. RESULTADOS	67
7. CONCLUSIONES	82
8. BIBLIOGRAFÍA	84

## INDICE DE CONTENIDO

1. RESUMEN	1
1.1. PALABRAS CLAVE	1
2. ABSTRACT	2
2.1. KEY WORDS	2
3. INTRODUCCIÓN	3
4. FUNDAMENTOS TEÓRICOS Y ESTADO DE ARTE	5
4.1. DISPOSITIVOS ELECTROENCEFALOGRAFICOS	5
4.1.1. Introducción	5
4.1.2. Funcionamiento de los dispositivos EEG	6
4.1.3. Sistema de representación	7
4.1.4. Señales EEG	10
4.1.5. Clasificación de los dispositivos EEG	11
4.1.6. MUSE 2	13
4.2. SISTEMA DE INTERFAZ CEREBRO-COMPUTADORA	14
4.2.1. Introducción	14
4.2.2. Historia	15
4.2.3. Clasificación de BCIs	16

4.2.3.1. Sistema Endógeno	16
4.2.3.1.1. Potenciales Corticales Lentos	16
4.2.3.1.2. Actividad Sensoriomotora	17
4.2.3.2. Sistema Exógeno	19
4.2.3.2.1. Potenciales Relacionados con Evento	19
4.2.3.2.2. Potencial Visual Evocado	20
4.2.4. Estructura de funcionamiento	21
4.2.4.1. Adquisición de la señal	22
4.2.4.1.1. Bluetooth de Baja Energía	23
4.2.4.1.2. Protocolo UDP	23
4.2.4.1.3. Protocolo OSC	24
4.2.4.2. Retirada de artefactos	26
4.2.4.3. Procesado de la señal	27
4.2.4.3.1. Mejora de la señal	27
4.2.4.3.2. Extracción de características	29
4.2.4.3.3. Reducción dimensional	32
4.2.4.4. Clasificación de las características	32
4.2.4.4.1. Regresión logística	33
4.2.4.4.2. XGboost	35
4.2.5. Aplicaciones	36
4.2.5.1. Interfaces Humano-Computadora (HCI)	37
4.2.5.2. Entretenimiento	38
4.2.5.3. Bienestar de las personas	38
4.3. MÉTRICAS PARA LA EVALUACIÓN DE UN MODELO	40
4.3.1. Introducción	40
4.3.2. Capacidad de Imaginación Motora del sujeto	40
4.3.3. Evaluación de los resultados del clasificador	41
4.3.3.1. Matriz de Confusión	41
4.3.3.2. Exactitud	42
4.3.3.3. Precisión	43
4.3.3.4. Exhaustividad	43
4.3.3.5. Valor F1	44
4.3.3.6. Coeficiente Kappa de Cohen	44
<b>5. MÉTODO OPERATIVO</b>	<b>46</b>
5.1. INTRODUCCIÓN	46
5.2. COMUNICACIÓN	47
5.3. LABVIEW	47
5.3.1. Adquisición de datos	47
5.3.2. Registro de datos	48
5.3.3. Estructura y funcionamiento del programa	50



5.3.3.1. Servidor para la adquisición de datos	50
5.3.3.2. Programa para el registro de datos	51
5.3.4. Ensayos para el registro	52
5.3.5. Archivos resultantes	53
5.4. PYTHON	55
5.4.1. Construcción de los datasets	55
5.4.2. Procesado de los datos	55
5.4.2.1. Preparación de los datos de entrenamiento	56
5.4.2.2. 1 <sup>er</sup> filtro FIR paso banda	56
5.4.2.3. Filtrado de artefactos	59
5.4.2.4. 2 <sup>o</sup> filtro FIR paso banda	60
5.4.2.5. Filtro CSP	62
5.4.3. Procesado del database de testeo	64
5.4.4. Clasificación de características	65
<b>6. RESULTADOS</b>	<b>67</b>
6.1. CAPACIDAD DE I.M. DEL SUJETO	67
6.2. EVALUACIÓN DE LOS CLASIFICADORES	68
6.2.1. Modelo de Regresión Logística	69
6.2.1.1. Valoración de la exactitud o "accuracy"	69
6.2.1.2. Matrices de Confusión	70
6.2.1.3. Precisión	72
6.2.1.4. Exhaustividad	72
6.2.1.5. Valor F1	72
6.2.1.6. Coeficiente Kappa de Cohen	73
6.2.2. Modelo de XGboost	74
6.2.2.1. Interpretación del porcentaje de precisión	74
6.2.2.2. Matrices de Confusión	75
6.2.2.3. Precisión	77
6.2.2.4. Exhaustividad	77
6.2.2.5. Valor F1	77
6.2.2.6. Coeficiente Kappa de Cohen	78
6.2.3. Comparación entre los clasificadores	78
6.3. DISCUSIÓN	79
6.3.1. Valoración del modelo	79
6.3.2. Hipótesis de mejora	80
6.3.2.1. Hipótesis para la mejora previa al procesado	80
6.3.2.2. Hipótesis para la mejora en el procesado	81
6.3.2.3. Hipótesis para la mejora tras el procesado	81

<b>7. CONCLUSIONES</b>	<b>82</b>
7.1. CONCLUSIONES GENERALES	82
7.2. LÍNEA FUTURA	82
<b>8. BIBLIOGRAFÍA</b>	<b>84</b>

## INDICE DE ILUSTRACIONES

<i>Ilustración 1: Tipos de montaje en dispositivos EEG</i>	7
<i>Ilustración 2: Sistema Internacional sin ubicar los electrodos</i>	8
<i>Ilustración 3: Sistema Internacional 10-10</i>	9
<i>Ilustración 4: Lóbulos que componen el cerebro</i>	9
<i>Ilustración 5: Sistemas de adquisición de datos disponibles en el mercado [1]</i>	12
<i>Ilustración 6: Representación de los electrodos de un dispositivo Muse en el Sistema Internacional [16]</i>	13
<i>Ilustración 7: Potenciales Corticales Lentos [18]</i>	17
<i>Ilustración 8: Áreas cerebrales de importancia</i>	18
<i>Ilustración 9: Actividad Sensoriomotora estrechamente correlacionada con la banda de frecuencia alpha [18]</i>	19
<i>Ilustración 10: Potencial Relacionado con Evento P300 [18]</i>	20
<i>Ilustración 11: Diagrama de bloques de un sistema BCI</i>	21
<i>Ilustración 12: Estructura de un mensaje en protocolo OSC [29]</i>	25
<i>Ilustración 13: Composición de un paquete OSC [29]</i>	25
<i>Ilustración 14: Lectura de una dirección OSC [29]</i>	26
<i>Ilustración 15: Regresión Logística Simple y Binaria [38]</i>	34
<i>Ilustración 16: Algoritmo de XGBoost [41]</i>	36
<i>Ilustración 17: Campos de investigación donde han contribuido las BCIs [1]</i>	37
<i>Ilustración 18: Diagrama del método operativo seguido</i>	46
<i>Ilustración 19: Interface de LabVIEW diseñada</i>	49
<i>Ilustración 20: Prueba ideada para la toma de datos</i>	52

<i>Ilustración 21: Identificación de un dataset.....</i>	<i>54</i>
<i>Ilustración 22: Señal EEG en bruto .....</i>	<i>57</i>
<i>Ilustración 23: Señal EEG del canal TP10.....</i>	<i>57</i>
<i>Ilustración 24: PSD de la señal en bruto .....</i>	<i>57</i>
<i>Ilustración 25: Coeficientes del filtro 1 .....</i>	<i>58</i>
<i>Ilustración 26: Respuesta del filtro 1 .....</i>	<i>58</i>
<i>Ilustración 27: Resultado en TP9 tras aplicar el filtro 1 .....</i>	<i>58</i>
<i>Ilustración 28: PSD tras aplicar el filtro 1.....</i>	<i>59</i>
<i>Ilustración 29: Señales EEG tras aplicar el filtro 1 .....</i>	<i>59</i>
<i>Ilustración 30: Correlación de la señal ante eventos de EOG .....</i>	<i>60</i>
<i>Ilustración 31: ICs .....</i>	<i>60</i>
<i>Ilustración 32: Resultados tras emplear el ICA .....</i>	<i>60</i>
<i>Ilustración 33: Señales EEG tras aplicar la técnica de ICA .....</i>	<i>61</i>
<i>Ilustración 34: Coeficientes del filtro 2 .....</i>	<i>62</i>
<i>Ilustración 35: Respuesta del filtro 2 .....</i>	<i>62</i>
<i>Ilustración 36: Resultado en TP9 tras aplicar el filtro 2 .....</i>	<i>62</i>
<i>Ilustración 37: Señal 1 al aplicar el CSP .....</i>	<i>63</i>
<i>Ilustración 38: Señal 2 al aplicar el CSP .....</i>	<i>63</i>
<i>Ilustración 39: Señal Resultante 1 (Train).....</i>	<i>64</i>
<i>Ilustración 40: Señal Resultante 2 (Train).....</i>	<i>64</i>
<i>Ilustración 41: Procesado de la Señal .....</i>	<i>64</i>
<i>Ilustración 42: Señal Resultante 1 (Test).....</i>	<i>65</i>
<i>Ilustración 43: Señal Resultante 2 (Test).....</i>	<i>65</i>
<i>Ilustración 44: Etiquetas totales para el entrenamiento .....</i>	<i>65</i>
<i>Ilustración 45: Etiquetas totales para el testeo .....</i>	<i>66</i>
<i>Ilustración 46: Evolución de la predicción con LR en función del tamaño de ventana usado en el alisado .....</i>	<i>69</i>
<i>Ilustración 47: Predicciones del modelo RL aplicado a los datos Train .....</i>	<i>71</i>

<i>Ilustración 48: Predicciones del modelo RL aplicado a los datos Test .....</i>	<i>71</i>
<i>Ilustración 49: Evolución de la predicción con XGboost en función del tamaño de ventana usado en el alisado .....</i>	<i>74</i>
<i>Ilustración 50: Predicciones del modelo XGboost aplicado a los datos Train .....</i>	<i>76</i>
<i>Ilustración 51: Predicciones del modelo XGboost aplicado a los datos Test.....</i>	<i>76</i>
<i>Ilustración 52: Diagrama UML de Actividad de la BCI planteada .....</i>	<i>83</i>

## INDICE DE TABLAS

<i>Tabla 1: Actividades cerebrales asociadas a cada ancho de banda .....</i>	<i>10</i>
<i>Tabla 2: Artefactos generados en las señales EEG más frecuentemente.....</i>	<i>11</i>
<i>Tabla 3: Estructura de la cabecera en un mensaje UDP [28] .....</i>	<i>24</i>
<i>Tabla 4: Técnicas para la extracción de características [34] .....</i>	<i>30</i>
<i>Tabla 5: Parámetros de una matriz de confusión .....</i>	<i>42</i>
<i>Tabla 6: Grado de acuerdo estimado en función del coeficiente k.....</i>	<i>45</i>
<i>Tabla 7: Información de interés sobre los datos transmitidos por Mind Monitor .....</i>	<i>48</i>
<i>Tabla 8: Etiquetas de identificación de los eventos en cada database .....</i>	<i>49</i>
<i>Tabla 9: Descripción del tipo de dato que posee cada columna en los databases .....</i>	<i>54</i>
<i>Tabla 10: Número de identificación de cada etiqueta.....</i>	<i>56</i>
<i>Tabla 11: Parámetros para el diseño el filtro 1 .....</i>	<i>58</i>
<i>Tabla 12: Parámetros para el diseño el filtro 2 .....</i>	<i>61</i>
<i>Tabla 13: Resultados del MIQ-R .....</i>	<i>67</i>
<i>Tabla 14: Exactitud de cada modelo al aplicarlo sobre Train y Test .....</i>	<i>68</i>
<i>Tabla 15: Matriz de confusión al aplicar el modelo RL sobre Train .....</i>	<i>70</i>
<i>Tabla 16: Matriz de confusión al aplicar el modelo RL sobre Test .....</i>	<i>70</i>
<i>Tabla 17: Matriz de confusión al aplicar el modelo XGboost sobre Train .....</i>	<i>75</i>
<i>Tabla 18: Matriz de confusión al aplicar el modelo XGboost sobre Test .....</i>	<i>75</i>
<i>Tabla 19: Resultados finales de cada modelo sobre los datos de testeo .....</i>	<i>79</i>

# 1. RESUMEN

A lo largo de la última década, la investigación de los patrones generados por los impulsos eléctricos procedentes del cerebro ha aumentado en gran medida. Este fenómeno se puede explicar debido a una mayor asequibilidad de los dispositivos electroencefalográficos (EEG) necesarios para captar dichas señales, así como una mejora de las técnicas para el procesado de las mismas. Por medio de este Trabajo de Fin de Grado, se pretende demostrar que, mediante de un dispositivo EEG comercial de cuatro canales de registro, se puede predecir pensamientos de imaginación motora. Éstos estarán orientados a establecer una comunicación humano-máquina con un sistema informático sin necesitar teclado, ratón o voz (Interface Cerebro-Computadora). Para ello, en primer lugar, se ha diseñado un software para la adquisición de datos, seguido de la realización de una serie de pruebas para crear una base de datos. Finalmente, estas señales han sido procesadas offline, construyendo varios modelos predictivos a través de algoritmos de clasificación. El resultado ha sido satisfactorio, logrando detectar ciertos patrones en las señales EEG gracias a técnicas como el Análisis de Componentes Independientes (ICA), los Patrones Espaciales Comunes (CSP) o la Regresión Logística (RL).

## 1.1. PALABRAS CLAVE

Señales Electroencefalográficas (EEG), Interfaz Cerebro-Computadora (BCI), MUSE 2, Imaginación Motora, extracción y clasificación de características.

## **2. ABSTRACT**

Along the last decade, the investigation of the patterns generated by the electric impulses from the brain has been highly increased. This phenomenon can be explained due to bigger affordability of the electroencephalographic (EEG) devices required to capture such signals, as well as an improvement in the techniques used to process them. Through this Final Degree Project, it is pretended to demonstrate that, with a four channel commercial EEG device, it is possible to predict motor imagery thoughts. They will be orientated to stablish a human-machine communication with an informatics system without needing keyboard, mouse or voice (Brain-Computer Interface). In order to achieve this, firstly, it has been designed a software for the data acquisition, followed by the realization of a test set to create a big database. Ultimately, these signals have been processed offline, building some predictive models by the use of classification algorithms. The result has been satisfactory, achieving the detection of some patterns in the EEG signals thanks to the use of techniques like the Independent Components Analysis (ICA), the Common Special Patterns (CSP) or the Logistic Regression (RL).

### **2.1. KEY WORDS**

Electroencephalographic Signals (EEG), Brain-Computer Interface (BCI), MUSE 2, Motor Imagery, feature extraction and classification.

### 3. INTRODUCCIÓN

En la actualidad, dentro del mercado el número de dispositivos EEG comerciales a la venta ha aumentado de forma considerable, sacrificando la calidad de la señal para hacerlos mucho más asequibles. Por otro lado, a lo largo de las últimas décadas, el aumento y optimización de las técnicas orientadas a procesar y predecir los impulsos eléctricos procedentes del cerebro ha sido considerable.

A causa de las razones citadas previamente, se está apreciando un incremento de investigaciones en torno al diseño de aplicaciones de Interfaz Cerebro-Computadora (BCI) que permitan a un usuario interactuar con una máquina por medio de sus ondas cerebrales. Entre otros, muchas universidades y empresas de Innovación y Desarrollo están viéndose interesadas en el estudio de este campo, al haberse vuelto mucho más accesible su estudio.

En consecuencia, muchos creen que en varias décadas el uso de este tipo de interfaces comenzará a verse generalizado dentro de la población, al igual que hoy en día ya estamos acostumbrados a ver implementados sistemas como el de reconocimiento facial o el de voz. De esta forma, la calidad de vida de las personas discapacitadas podría verse ampliamente mejorada, pudiendo equiparse, por ejemplo, de brazos prostéticos, así como comunicarse con máquinas, como una silla de ruedas o una computadora, sin requerir de sus extremidades.

Con este ánimo de progreso, mediante el presente Trabajo de Fin de Grado, se pretende dar apoyo al “*Instituto Iberoamericana de Innovación (I3b)*” llevando a cabo un estudio de viabilidad sobre el desarrollo de una BCI orientada al control de un ordenador sin precisar de ratón.

En primer lugar, se va a realizar una evaluación del estado de arte en relación al campo de las señales electroencefalográficas y al diseño de sistemas BCI, recopilando la información contenida en diversos estudios.

Posteriormente, haciendo uso del software *LabVIEW*, se va a realizar el diseño de un programa para la adquisición de datos orientado al registro de las señales EEG en bruto y su almacenamiento en *databases*.

Los ensayos se harán mediante un dispositivo comercial de cuatro canales de registro denominado “*MUSE 2*”. Cedido por la empresa “*I3b*”, el instrumento, que se encuentra a la venta en el mercado a un precio asequible, se podrá comunicar con un dispositivo móvil a través de Bluetooth de Baja Energía. Éste, funcionando a modo de

## Introducción

puente, será capaz de transmitir los datos al ordenador gracias al protocolo conocido como *UDP*, registrándolos dentro del programa mencionado previamente.

Tras la planificación de un procedimiento para realizar ensayos, se hará una serie de grabaciones de las señales captadas por los electrodos del dispositivo *MUSE 2*. Los impulsos, correctamente etiquetados en función de ciertos pensamientos de *imaginación motora*, se almacenarán en distintos *datasets*, los cuales serán agrupados, dejando solamente un par de archivos. Éstos serán la base de datos de entrenamiento, a partir de la cual construir modelos para la clasificación binaria de varios eventos, y la de testeo, orientada a probar su capacidad de predicción.

Encaminados al control de un puntero, los pensamientos a clasificar, en un inicio, iban a ser cuatro parejas:

- Imaginación Motora de desplazamiento de la mano diestra hacia la izquierda o derecha.
- Imaginación Motora de desplazamiento de la mano diestra hacia arriba o abajo.
- Imaginación Motora de movimiento horizontal de mano o inacción.
- Imaginación Motora de movimiento vertical de mano o inacción.

Sin embargo, por problemas de tiempo, en el presente Trabajo de Fin de Grado, únicamente se va a clasificar la primera pareja de etiquetas citada, realizando la prueba a un único sujeto.

A través del lenguaje de programación *Python* y por medio del entorno de *Google Colab*, se va a realizar tanto el procesado de la señal en bruto como el entrenamiento y testeo de varios modelos, hechos a partir de diferentes algoritmos de clasificación.

Por último, para concluir el proyecto, se va a realizar un análisis extendido acerca de los datos obtenidos mediante las diferentes técnicas. También se va a proponer hipótesis para la mejora de los resultados, así como los puntos en los que se debería dar continuidad a la investigación.



## 4. FUNDAMENTOS TEÓRICOS Y ESTADO DE ARTE

### 4.1. DISPOSITIVOS ELECTROENCEFALOGRÁFICOS

#### 4.1.1. Introducción

Dentro de las células, la neurona es el elemento principal del sistema nervioso, y su misión es la recepción, procesamiento y emisión de la información captada por nuestro cuerpo. Para lograrlo, a lo largo de nuestro cerebro se extiende una amplia red neuronal capaz de recibir e interpretar los datos del entorno, así como de mandar órdenes de forma rápida y continuada a todo nuestro cuerpo.

Dicha información se transmite por medio de señales tanto químicas como eléctricas, pudiendo ser sus patrones estudiados e interpretados.

A lo largo de la historia reciente, se ha avanzado en gran medida en el campo de la detección de los impulsos eléctricos con origen en el cerebro. Hoy en día hay tal variedad de técnicas, que ha obligado a establecer una separación entre los distintos dispositivos, dividiéndolos en tres grupos: *Invasivos*, *Parcialmente Invasivos* y *No Invasivos* [1]. La característica principal que diferencia cada uno de los métodos es la posición del sensor encargado de la lectura de las señales, siendo el no invasivo el único que no establece un contacto directo con la corteza cerebral.

Un ejemplo conocido de dispositivo invasivo, es el caso del Electrocorticograma (ECoG) que, al introducirse por medio de una cirugía, entra en contacto directo con el cerebro, sorteando así el cráneo y el cuello cabelludo. De este modo, es capaz de abarcar un mayor rango de frecuencias con un nivel de detalle muy amplio.

El buen funcionamiento de los *dispositivos no invasivos* ha sido demostrado en varias ocasiones para, por ejemplo, controlar prótesis o tratar la depresión [2]. A su vez, a lo largo de los últimos años se está hablando de avances de gran importancia dentro de este campo, siendo la presentación en Septiembre de 2020 del proyecto "*Neuralink*" [2] [3] una manifestación de ello. Sin embargo, la implementación ocasional de dispositivos basados en este tipo de tecnologías sigue necesitando mucha inversión y años de desarrollo.

Por otro lado, al prescindir de una intervención quirúrgica, así como al ser mucho menos complejos y, por lo tanto, más asequibles económicamente, los aparatos no invasivos son el grupo que ha sido más estudiado a lo largo de muchas décadas. En

consecuencia, se ha experimentado un notable aumento tanto en sus parámetros de medición como en la calidad de señal y fiabilidad, siendo habitualmente utilizados en, por ejemplo, la detección y diagnóstico de ciertas enfermedades en el ámbito sanitario.

Dentro del rango de dispositivos no invasivos, éstos se pueden agrupar en grupos diferentes según la técnica utilizada, destacando entre otros las Imágenes de Resonancia Magnética (fMRI) y el Electroencefalograma [1].

- Las fMRIs funcionan al detectar la variación de oxígeno en sangre a causa de una actividad neuronal, siendo una técnica muy útil para evaluar el estado de salud del cerebro. En comparación con los dispositivos EEG, destaca por tener una resolución espacial muy elevada, frente a su resolución temporal, que es bastante reducida [4].
- El EEG mide los impulsos eléctricos producidos en el cerebro, estando la técnica implementada en la actualidad en centros sanitarios para la detección de enfermedades como epilepsia o tumores, así como trastornos del sueño. La resolución temporal de dispositivos de este tipo es muy alta, permitiendo visualizar la evolución de las señales a tiempo real. Por el contrario, tienen una resolución espacial limitada [4] a los puntos en los que los electrodos son colocados.

A su vez, otras técnicas no invasivas alternativas utilizadas para poder visualizar la actividad cerebral son, por ejemplo, la Tomografía por Emisión de Positrones (PET) y la Magnetoencefalografía (MEG) [5].

Uno de los problemas principales que presentan tanto las fMRIs como la PET y MEG es el requerimiento de un equipo e instalaciones de coste elevado [5]. Es por ello que los dispositivos del tipo EEG suelen resultar más sencillos de adquirir, siendo uno de ellos el que se va a utilizar en el proyecto actual.

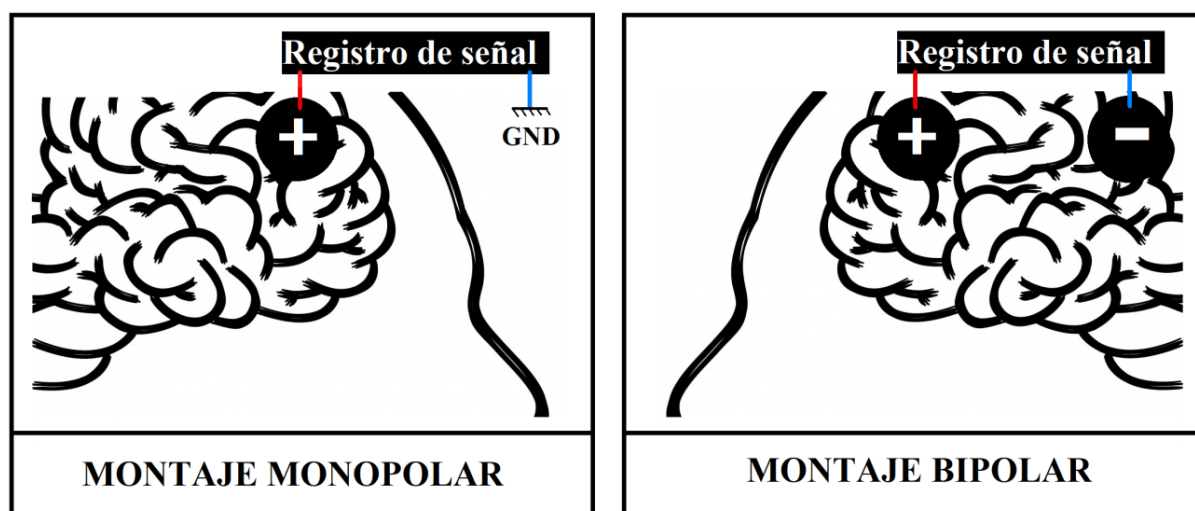
#### ***4.1.2. Funcionamiento de los dispositivos EEG***

El término “Electroencefalograma” comenzó a ser utilizado por el neurólogo y psiquiatra alemán Hans Berger en el año 1929 cuando se refería al registro de las oscilaciones eléctricas generadas a lo largo de la corteza cerebral [6].

La técnica consiste en la fijación de un conjunto de electrodos al cuello cabelludo (electroencefalograma estándar [6]), funcionando como sensores al captar la diferencia de potencial de la señal eléctrica producida por las neuronas. Posteriormente, los datos son amplificados y, por último, registrados para poder realizar un análisis.

Para lograr su objetivo, los dispositivos EEG se encuentran formados por un conjunto de derivaciones que reciben el nombre de canales de registro, que, por ejemplo, en las aplicaciones sanitarias actuales, llegan a ser de entre 16 y 24 [6].

El conjunto total de todas las derivaciones recibe el nombre de montaje. Según la colocación de los electrodos, éste puede ser clasificado como bipolar o monopolar. Dicha denominación dependerá de si el punto en el cual se sitúa el electrodo de referencia posee actividad cerebral o no [6], respectivamente (*Ilustración 1*).

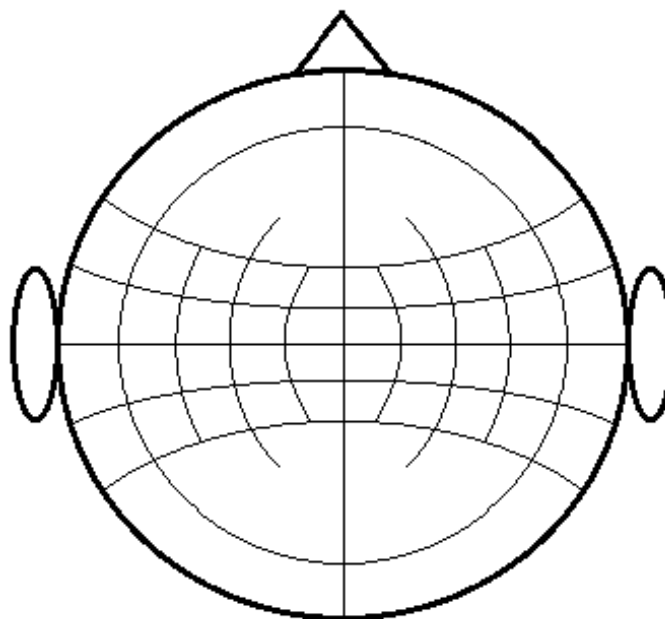


***Ilustración 1: Tipos de montaje en dispositivos EEG***

En el caso de un dispositivo monopolar, éste se ubicará en una zona en la cual la tensión eléctrica tienda a 0 V, como por ejemplo el lóbulo de la oreja. De esta forma, el valor de los impulsos eléctricos captado será únicamente el detectado por el electrodo activo. En cambio, si el montaje es bipolar, tal y como se ha comentado, el resultado final será la diferencia entre los dos electrodos.

### ***4.1.3. Sistema de representación***

Pese a que hay un gran número de formas de representación, la distribución de los canales en los dispositivos EEG se suele mostrar mediante un diagrama como el que se puede ver en la *Ilustración 2*. Éste es conocido como el *Sistema Internacional* [4], y, sobre él se identifica con un círculo las posiciones más importantes en las que se coloca el conjunto de los electrodos, sombreando las que hace uso el montaje del cual se esté hablando.



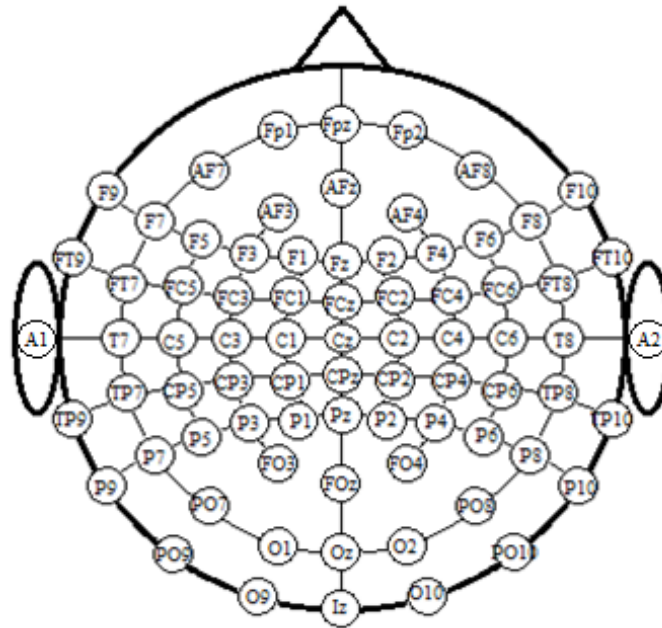
**Ilustración 2: Sistema Internacional sin ubicar los electrodos**

El sistema fue ideado por la Federación Internacional de Neurofisiología Clínica (IFCN) [7] con objeto de encontrar una forma estandarizada para representar los montajes EEG [4].

Originalmente éste era conocido como el protocolo de ubicación de electrodos "10-20", a través del cual se podía colocar un total de 21 sensores a lo largo de la cabeza del sujeto, usando como puntos *de referencia* el *nasion*, el *inion* y los "*puntos preauriculares*" [4]. Cabe destacar que, hoy en día, el uso de este sistema está muy extendido dentro del entorno sanitario.

Sin embargo, el sistema 10-20 no contempla el uso de un mayor número de sensores, quedándose escaso en investigaciones de alto presupuesto. Además, se ve muy limitado en torno a los puntos donde colocar estos electrodos, solo pudiendo escoger entre 21 posibilidades. Por esa razón, en 1985 el *Sistema Internacional* fue extendido [4], creando un protocolo a través del cual se puede colocar hasta 74 electrodos. Éste es muy empleado hoy en día dentro de investigaciones, siendo conocido como sistema "10-10" (*Ilustración 3*).

Para conocer de forma normalizada el lugar que ocupa cada electrodo en ambos sistemas, a éstos se les asigna una o varias letras, así como un número, haciendo variar su combinación en función del lugar de la cabeza que ocupe.

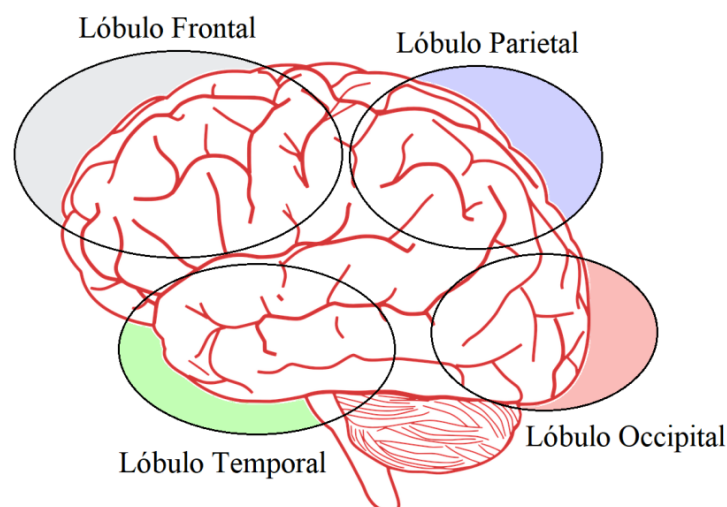


**Ilustración 3: Sistema Internacional 10-10**

Por un lado, el electrodo es identificado en función de la zona de actividad cerebral (lóbulo) donde se coloca, añadiendo una letra mayúscula con la respectiva abreviatura de dicha área. En caso de que el electrodo se ubique en una zona intermedia, se designa con las dos letras correspondientes a las zonas entre las cuales se sitúa.

Las zonas de actividad cerebral de las cuales se está hablando son las que se puede visualizar en la *Ilustración 4*.

A su vez, la letra C es la encargada de identificar los electrodos posicionados en la línea que atraviesa el centro verticalmente.



**Ilustración 4: Lóbulos que componen el cerebro**

Por otro lado, el número identificativo aumentará a medida que el electrodo vaya ocupando una posición más extrema, siendo pares el conjunto de electrodos ubicados en el lado derecho de la cabeza e impares los del izquierdo.

#### 4.1.4. Señales EEG

Las señales captadas por los canales de un dispositivo electroencefalográfico comparten ciertas características entre todas.

En primer lugar, el valor de la tensión real que detecta un electrodo es ínfimo, siendo ésta representada en el orden de los microvoltios ( $\mu V$ ).

Sin embargo, generalmente, a la señal original se le suma el ruido causado, por ejemplo, por la corriente eléctrica alterna del entorno (señal sinusoidal de 50 Hz de frecuencia), o por el contacto entre electrodo y cuello cabelludo (normalmente con contacto en seco). En consecuencia, es generada una tensión de *offset* que modifica en gran medida los datos originales.

Por otro lado, las señales no poseen un periodo fijo, oscilando arrítmicamente y dejando formas de ondas totalmente irregulares. Por ello, para lograr una visualización clara de la evolución de la señal, esta suele ser dividida entre las diferentes bandas de frecuencia asociadas a la actividad neuronal. Cada una de ellas está relacionada con un tipo de actividad mental diferente [8], tal y como podemos ver en la *Tabla 1*.

Nombre de banda	Ancho de banda	Actividad cerebral con la que se relaciona
<b>Delta</b>	1 – 4 Hz	Estado de sueño profundo
<b>Theta</b>	4 – 8 Hz	Relajación y meditación profunda
<b>Alpha</b>	7,5 – 13 Hz	Estado meditativo, creativo o imaginativo
<b>Beta</b>	13 – 30 Hz	Concentración, estrés o excitación
<b>Gamma</b>	30 – 44 Hz	Estado de procesamiento de información de alto nivel

**Tabla 1: Actividades cerebrales asociadas a cada ancho de banda**

Otro rasgo característico de las señales EEG son los llamados *artefactos*, componentes ajenos a la actividad neuronal que interfieren cuando ocurren ciertos eventos, causando alteraciones [9]. En la *Tabla 2* se ha tratado de resumir los más conocidos e importantes.

Tal y como se puede ver, en ocasiones es imposible evitar la generación de artefactos, los cuales, pese a que en ocasiones son patrones de gran utilidad, la mayoría de las veces resultan un inconveniente si se quiere examinar una señal.

Artefacto	Causas	¿Es Evitable?
<b>Cardiaco (ECG)</b>	Es causado por el latir de corazón cuando un electrodo se coloca en una zona donde se pueda detectar.	A menos que se evite ciertas posiciones del electrodo, no es posible de evitar.
<b>Muscular (EMG)</b>	Causado por movimientos musculares, tanto faciales como de extremidades.	Se pueden minimizar en gran medida, evitando mover ninguna parte del cuerpo mientras se está registrando datos.
<b>Ocular (EOG)</b>	Causado durante los parpadeos, así como cuando se da un movimiento del globo ocular.	Se puede evitar el movimiento ocular, pero el parpadeo a veces resulta inevitable.

**Tabla 2: Artefactos generados en las señales EEG más frecuentemente**

Para solucionar el problema, existen algoritmos para detectar dichos eventos, pudiendo eliminar los tramos concretos en los que ocurren. Sin embargo, para las aplicaciones a tiempo real, es recomendable una retirada del componente de la señal que se asocie a un artefacto. Para ello, hoy en día se han desarrollado algoritmos como los que emplean la técnica de ICA (Independent Component Analysis) [10], los cuales serán tratados dentro de apartados posteriores.

#### 4.1.5. Clasificación de los dispositivos EEG

En la actualidad la innovación a la hora de diseñar dispositivos EEG ha llegado hasta tal punto que podemos llegar a diferenciarlos en dos grandes grupos en función del número de canales.

En primer lugar, se encuentran los del *grado de investigación*, aparatos complejos que cuentan con un gran número de electrodos (hasta 64, aproximadamente) [11] y que, por consiguiente, aportan una información elevada y de calidad.

Generalmente éstos son usados dentro de investigaciones de alto presupuesto, aplicando sobre el paciente un gel conductor con objeto de minimizar el ruido de la señal captada. A su vez, cabe destacar su aplicación dentro del sector sanitario, permitiendo la detección de ciertas enfermedades como la epilepsia.

Pese a la antigüedad de los dispositivos EEG del *grado de investigación* (se emplean desde el siglo pasado), salvo en hospitales y en ciertos laboratorios, hasta la actualidad no han podido ser implementados en nuestro día a día. El motivo puede deberse a varias causas, como la portabilidad nula que tienen, lo cual dificulta su uso para aplicaciones reales, o el tiempo de preparación necesario a la hora de posicionar los electrodos sobre el sujeto.



## Fundamentos Teóricos y Estado de Arte

Por otro lado, tal y como se puede apreciar en algunos estudios [12], ciertos investigadores ya durante la primera década del siglo XXI criticaban el elevado coste que un dispositivo de este tipo podía suponer. En la actualidad, el precio para adquirirlo no ha variado notablemente, pudiendo oscilar entre los 20000 y 250000 dólares [13].

A modo de respuesta, a lo largo de los últimos años, poco a poco se ha ido asentando otro grupo de dispositivos EEG mucho más accesibles para la población en general, los del *grado de consumo*.

Su desarrollo ha sido posible gracias a la labor realizada por numerosas empresas, que han visto en el campo de los instrumentos *no invasivos* una oportunidad de negocio. Entre otras, han surgido compañías como Qneuro, Muse, NeuroSky o EMOTIV, así como algunas empresas famosas como, por ejemplo, Facebook, han decidido invertir recursos para el desarrollo de aplicaciones [2].

En los dispositivos del *grado de consumo*, el número de electrodos es reducido de una forma considerable, resultando en un número que varía entre 1 y 14, de forma aproximada [11] y, en consecuencia, abaratando el coste de los mismos. A su vez, con objeto de aportar una mayor comodidad de uso, los sensores se encuentran incorporados dentro del dispositivo, realizando un contacto en seco con la superficie de la cabeza, es decir, prescindiendo de la aplicación de gel conductor.

Además del bajo número de canales de registro, estos dispositivos son mucho más simples, con un diseño portable y orientado a su uso cotidiano en manos de un usuario con menor experiencia. En ocasiones pueden ser vendidos con el objeto de ofrecer unos servicios determinados, como, por ejemplo, la detección de ciertos estados mentales.

Brand	Product	Wearable	Sensors type	Channels No.	Locations	Sampling rate	Transmission	Weight
NeuroSky	MindWave	Yes	Dry	1	F	500 Hz	Bluetooth	90g
Emotiv	EPOC(+)	Yes	Dry	5-14	F, C, T, P, O	500 Hz	Bluetooth	125g
Muse	Muse 2	Yes	Dry	4-7	F, T		Bluetooth	
OpenBCI	EEG Electrode Cap Kit	Yes	Wet	8-21	F, C, T, P, O		Cable	
Wearable Sensing	DSI 24; NeuSenW	Yes	Wet; Dry	7-21	F, C, T, P, O	300/600 Hz	Bluetooth	600g
ANT Neuro	eego mylab / eego sports	Yes	Dry	32 - 256	F, C, T, P, O	Up to 16 kHz	Wi-Fi	500g
Neuroelectrics	STARSTIM; ENOBIO	Yes	Dry	8-32	F, C, T, P, O	125-500 Hz	Wi-Fi; USB	
G.tec	g.NAUTILUS series	Yes	Dry	8-64	F, C, T, P, O	500 Hz	Wireless	140g
Advanced Brain Monitoring	B-Alert	Yes	Dry	10-24	F, C, T, P, O	256Hz	Bluetooth	110g
Cognionics	Quick	Yes	Dry	8-30; (64-128)	F, C, T, P, O	250/500/1k/2k Hz	Bluetooth	610g
mBrainTrain	Smarting	Yes	Wet	24	F, C, T, P, O	250-500 Hz	Bluetooth	60g
Brain Products	LiveAmp	Yes	Dry	8-64	F, C, T, P, O	250/500/1k Hz	Wireless	30g
Brain Products	AntiCHamp	Yes	Dry	32-160	F, C, T, P, O	10k Hz	Wireless	1.1kg
BioSemi	ActiveTwo	No	Wet (Gel)	280	F, C, T, P, O	2k/4k/8k/16k Hz	Cable	1.1kg
EGI	GES 400	No	dry	32-256	F, C, T, P, O	8k Hz	Cable	
Compumedics Neuroscan	Quick-Cap + Graef 4k	No	Wet	32-256	F, C, T, P, O		Cable	
Mitsar	Smart BCI EEG Headset	Yes	Wet	24-32	F, C, T, P, O	2k Hz	Bluetooth	50g
Mindo	Mindo series	Yes	Dry	4-64	F, C, T, P, O		Wireless	

Abbreviation: Frontal (F), Central (C), Temporal (T), Partial (P), and Occipital (O)

### Ilustración 5: Sistemas de adquisición de datos disponibles en el mercado [1]



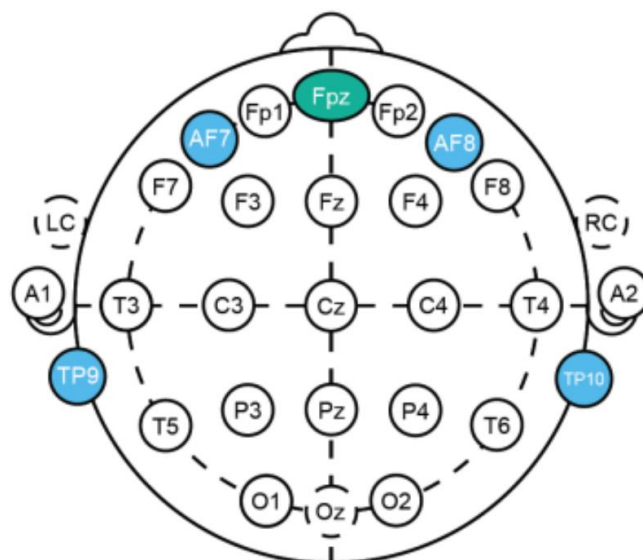
En conclusión, los dispositivos del *grado de consumo* priorizan la portabilidad, la comodidad a la hora de utilizarse y el precio, a costa de perder calidad en la señal captada. Sin embargo, a lo largo de estos últimos años, se ha logrado demostrar que, en estudios de menor presupuesto, con dispositivos de estas características, es posible obtener un resultado parecido al logrado con material del *grado de investigación* [14]. Para ello, únicamente se debe emplear los canales de registro adecuados, así como el uso de las técnicas correctas.

En la *Ilustración 5* se puede visualizar una tabla que agrupa un amplio conjunto de dispositivos EEG disponibles en la actualidad. Dentro del presente proyecto, se va a hacer uso de un *MUSE 2*, del cual se va a proceder a hablar a continuación.

#### 4.1.6. MUSE 2

El conjunto de dispositivos electroencefalográficos MUSE [15] son un grupo de productos del *grado de consumo* disponibles en el mercado y diseñados para poder usarse a modo de complemento por practicantes de meditación.

La labor que éste desempeña es la detección del nivel de relajación mental del usuario, devolviendo una realimentación en forma de diferentes sonidos, que variarán en función de lo bien que se esté llevando a cabo la sesión. Éstos generalmente son escuchados a través de unos auriculares.



**Ilustración 6: Representación de los electrodos de un dispositivo Muse en el Sistema Internacional [16]**

Su funcionamiento se basa en cuatro sensores calibrados y distribuidos a lo largo del dispositivo, el cual tiene forma de cinta. Tal y como se puede ver en la *Ilustración 6*,

dos de éstos (*AF7* y *AF8*) se ubican en la frente y los otros dos (*TP9* y *TP10*) detrás de las orejas, compartiendo todos el mismo electrodo *de referencia* (*Fpz*).

Una vez colocado el MUSE, los sensores comienzan a funcionar como electrodos, detectando los impulsos cerebrales que puedan ser percibidos dentro de sus respectivas posiciones.

Por otro lado, la transmisión de los datos es posible gracias a una comunicación a través del *Bluetooth de Baja Energía (BLE)* y, gracias a una aplicación diseñada por la propia compañía, procesados dentro de un dispositivo móvil para poder predecir el estado mental del usuario.

Pese a que el uso habitual de este instrumento no es la investigación, una característica que ha reforzado, en gran medida, a esta serie de productos, ha sido su facilidad para ser empleados para dicho fin. Inicialmente se distribuyeron softwares que facilitaban en gran medida tanto la adquisición, como el procesamiento de las señales en bruto. En la actualidad, pese a que el acceso a estos programas se haya dificultado debido a decisiones de la compañía, hay usuarios que han diseñado sus propias aplicaciones. Un ejemplo sería la aplicación "*Mind Monitor*" [17].

El conjunto de dispositivos MUSE ha ido evolucionando a lo largo de la segunda mitad de esta última década. MUSE 2, por ejemplo, es uno de los últimos modelos puestos en venta, el cual tiene incorporado un acelerómetro, un giroscopio, y, en caso de que se precisara, es capaz de medir el pulso cardíaco.

## 4.2. SISTEMA DE INTERFAZ CEREBRO-COMPUTADORA

### 4.2.1. Introducción

El concepto de Interfaz Cerebro-Computadora (BCI), se refiere a cualquier sistema de comunicación directa entre cerebro y un dispositivo externo con objeto de ejercer un control sobre este último.

A raíz del aumento de dispositivos de lectura de las reacciones neuronales, así como un mayor conocimiento de las mismas, en la actualidad se está viviendo un auge en torno a la investigación de este tipo de tecnologías.

Al igual que en el caso de los dispositivos, en función de la posición que tenga el instrumento empleado para registrar la información cerebral, la interface puede ser considerada invasiva o no invasiva [2].

En este trabajo se va a hablar únicamente acerca de los sistemas BCI no invasivos que hacen uso de un dispositivo Electroencefalográfico.

### 4.2.2. *Historia*

Partiendo de la idea que tuvo Hans Berger en 1929, durante los siguientes setenta años se fue asentando el aprovechamiento de la técnica EEG para detectar trastornos neurológicos, así como para investigar el funcionamiento del cerebro. También, en ciertas ocasiones, se había llegado a explorar positivamente la idea de aprovechar este tipo de dispositivos para tareas terapéuticas [18].

En cambio, pese a que en esos tiempos siempre se especuló sobre la posibilidad de poder utilizar las señales captadas para traducir ciertas órdenes directamente desde el cerebro, la idea fue poco tomada en serio a causa de las siguientes razones [18]:

- La fiabilidad y resolución de las señales EEG estaban limitadas a causa de la complejidad del cerebro, traduciéndose en la incapacidad de detectar un patrón en la respuesta ante un estímulo.
- No había tecnología capaz de analizar señales EEG en línea o ésta era muy costosa.
- Las posibilidades de comunicación que podía ofrecer un sistema de este tipo estaban, en esa época, muy limitadas.

Sin embargo, durante la última década del siglo XX esta tendencia pesimista comenzó a revertirse, explicándose principalmente a causa de la revolución tecnológica del momento (auge de los ordenadores).

A raíz de esta situación, se comenzaron a realizar varios estudios gracias a los cuales se detectó cierta correlación en algunos parámetros de la señal EEG cuando tenía lugar un evento [18]. De este modo se comenzó a asentar el concepto de BCI, el cual poco a poco fue cogiendo popularidad entre los investigadores.

En la actualidad, gracias a la tecnología y los medios disponibles, el conocimiento que se tiene sobre los dispositivos EEG y su aplicación en sistemas BCI nada tiene que ver con el que había a principios de siglo, habiendo experimentado un notable desarrollo.

A causa de este avance, actualmente, algunas empresas ya han conseguido poner a la venta sus propias aplicaciones de forma rentable económicamente. Un ejemplo podría ser, por ejemplo, la aplicación principal de MUSE 2 [15], de la que ya se ha hablado previamente y, gracias a la cual, se puede observar el nivel de concentración de un sujeto durante una práctica de meditación.

A parte de las aplicaciones que ya están siendo comercializadas, hay múltiples en estudio al mismo tiempo que se innova en las técnicas, explorándose así posibilidades nuevas que se podrían ver en el futuro.

### 4.2.3. Clasificación de BCIs

Toda aplicación que busque utilizar las señales EEG procedentes del cerebro de un sujeto con objeto de controlar un dispositivo, o parte de éste, contaría como sistema BCI. Sin embargo, entre éstos hay diferencias notables por las cuales, generalmente, suelen ser divididos en varios grupos.

En el presente trabajo se han establecido dos grandes categorías de sistemas BCI a partir de las cuales éstos pueden ser divididos en varios subgrupos. Si una interface de este tipo emplea más de uno de los subgrupos, ésta se consideraría que está basada en *Múltiples Neuromecanismos (MNs)* [19].

Según un artículo publicado en los cuadernos de la Fundación General CSIC [5], una señal EEG puede tener diferentes naturalezas en función del papel que ocupa el usuario en el control de la interface:

- Endógena.
- Exógena.

#### 4.2.3.1. Sistema Endógeno

Un sistema BCI endógeno funciona mediante el control directo del usuario que lo está utilizando, dependiendo su buen funcionamiento de que éste sea capaz de controlar la actividad neurofisiológica del cerebro para generar un patrón. Es por esta dependencia que están caracterizados por necesitar un entrenamiento más intensivo para lograr que puedan funcionar correctamente [5].

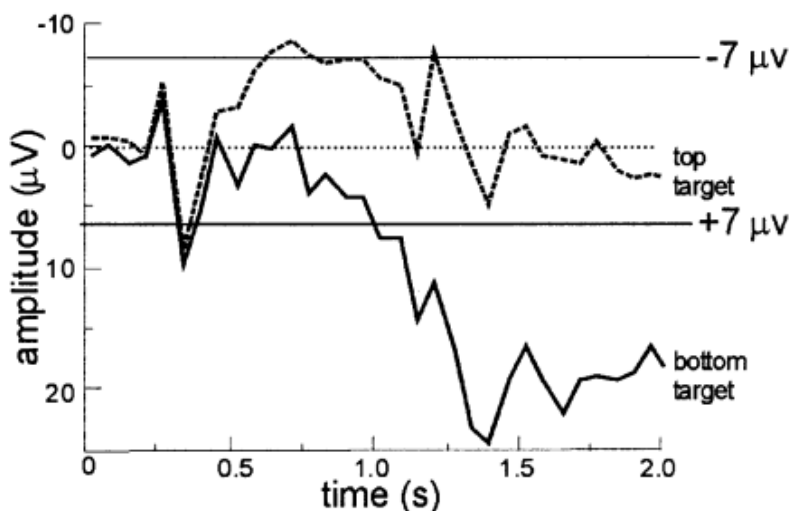
Dos ejemplos de BCI de este tipo son las basadas en los Potenciales Corticales Lentos (SCP) y en la actividad sensoriomotora [5].

A parte, cabe destacar otras técnicas de las cuales no se va a entrar a detalle en el presente trabajo. Entre otras, se debe mencionar la actividad que presentan las células neuronales (ANC) al realizar ciertos movimientos o, por ejemplo, la respuesta a distintas tareas mentales como resolver problemas matemáticos o contar mentalmente [19].

##### 4.2.3.1.1. Potenciales Corticales Lentos

Los *Potenciales Corticales Lentos* son alteraciones del valor de la tensión, producidas sobre la corteza cerebral (*Ilustración 7*) y presentando una duración

aproximada de entre 0,5 y 10 segundos [5]. Los SCP de valores negativos son generalmente asociados al movimiento, así como a otras actividades relacionadas con una elevada actividad cortical. Sin embargo, los valores positivos se atribuyen a una excitación reducida, como la que se presenta durante el sueño profundo [18].



**Ilustración 7: Potenciales Corticales Lentos [18]**

Un gran número de investigaciones dentro del campo de los SCP han sido atribuidas a Birbaumer, quien, a lo largo de 30 años, se dedicó a estudiar este tipo de respuestas cerebrales [18]. De este modo, se logró demostrar que estas respuestas pueden ser controladas por el usuario a base de aprendizaje, llegando en junio del 2000, junto con un grupo de compañeros, a diseñar una BCI que denominaron *Dispositivo de Traslación de Pensamientos (TTP)* [20]. A través de la misma, se logró que una serie de personas con ALS, tras un largo periodo de entrenamiento (190 sesiones el más rápido), pudiera comunicarse por escrito con otros individuos de su entorno.

#### 4.2.3.1.2. Actividad Sensoriomotora

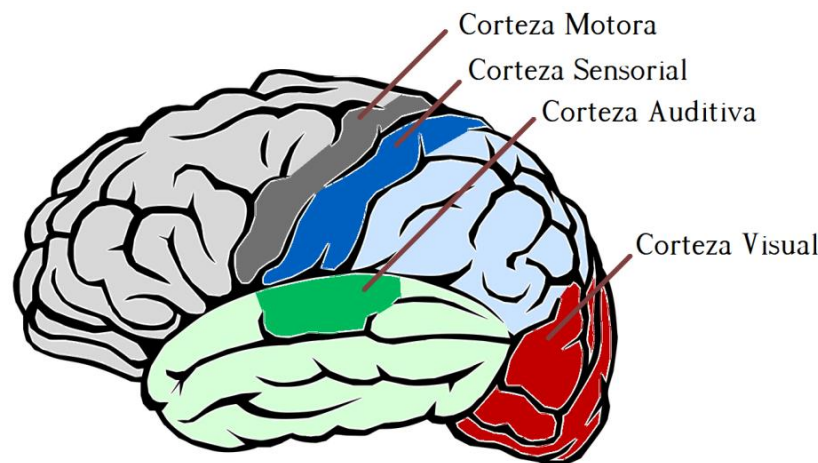
Entre el lóbulo frontal y el lóbulo parietal del cerebro se encuentra la corteza motora y la sensorial (*Ilustración 8*), teniendo la actividad de esta área la función principal de gestionar el sistema motor.

Los impulsos eléctricos generados en la corteza sensoriomotora se encuentran relacionados estrechamente con las bandas de frecuencia alpha (denominada en muchos proyectos como “mu”) y beta [18].

Por ello, los investigadores han tenido facilidad a la hora de realizar estudios sobre la búsqueda de patrones relacionados con la actividad motora en señales EEG, ya que el espectro se queda bastante reducido. A través de la experiencia y la observación se ha

logrado demostrar que, cuando un movimiento es producido o imaginado, se produce un cambio en la amplitud dentro de las bandas mencionadas de forma previa (*Ilustración 9*). Generalmente, en esos casos, el espectro experimenta una disminución de su valor (ERD) o una desincronización (ERS) [5].

- Los ERD son ocasionados cuando tiene lugar un movimiento o el usuario se está preparando para ejecutarlo, haciendo que los valores de las frecuencias en alpha y beta decrezcan.
- Los ERS, por el contrario, aparecen cuando se finaliza un movimiento o al producirse una relajación. En esta situación se ha observado que la amplitud en las bandas aumenta.

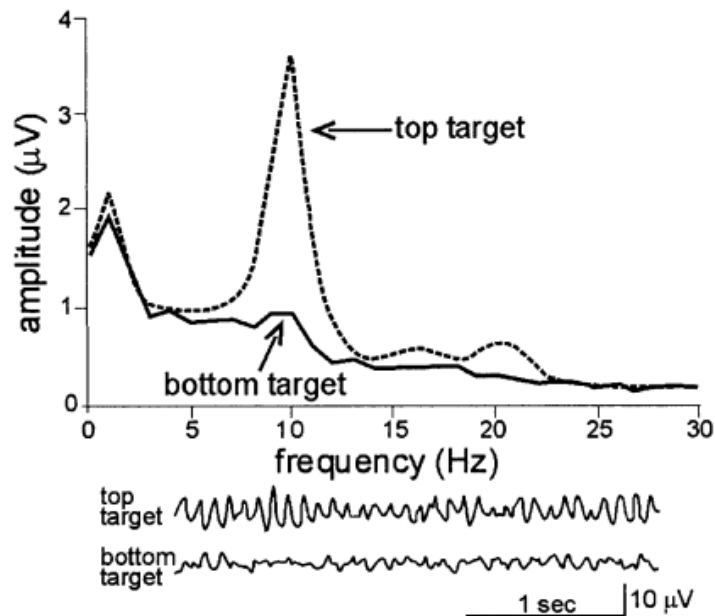


***Ilustración 8: Áreas cerebrales de importancia***

Tal y como se ha mencionado previamente, estos fenómenos ocurren tanto cuando se realizan en la realidad como cuando se imaginan, siempre teniendo en cuenta que algunos usuarios tienen una mayor capacidad de imaginación motora que otros.

Dentro de este ámbito cabe destacar a investigadores como Pfurtscheller, quien se dedicó a estudiar los patrones de las señales EEG generados a causa de imaginación motora. De este modo, él y su equipo llegaron a plantear, a principios de siglo, una BCI capaz de distinguir entre diferentes acciones, como un movimiento de la mano derecha o, por el contrario izquierda [18].

Por último, dentro de la categoría de BCIs basados en la actividad sensoriomotora, el sistema puede emplear otro tipo de respuestas a parte de la variación de frecuencia en las bandas alpha y beta. En el presente trabajo se va a mencionar, por ejemplo, los Potenciales Relacionados con Movimiento (MRPs), que son variaciones en la señal que comienzan entre 1 y 1.5 segundos antes de iniciar un movimiento [19].



**Ilustración 9: Actividad Sensoriomotora estrechamente correlacionada con la banda de frecuencia alpha [18]**

#### 4.2.3.2. Sistema Exógeno

Un sistema exógeno está caracterizado por utilizar potenciales evocados producidos en las señales EEG para poder dar órdenes a la interface.

Los *potenciales evocados* son reacciones que pueden ser visualizadas en una señal EEG cuando tiene lugar un estímulo externo. Este tipo de señales tienen un patrón bien definido en el dominio del tiempo, prescindiendo de un entrenamiento intensivo, a diferencia de las BCIs endógenas [5].

Los dos tipos de potenciales evocados más utilizados en investigaciones son los *Potenciales Relacionados con Evento* (ERP) y el *Visual Evocado* (VEP).

##### 4.2.3.2.1. Potenciales Relacionados con Evento

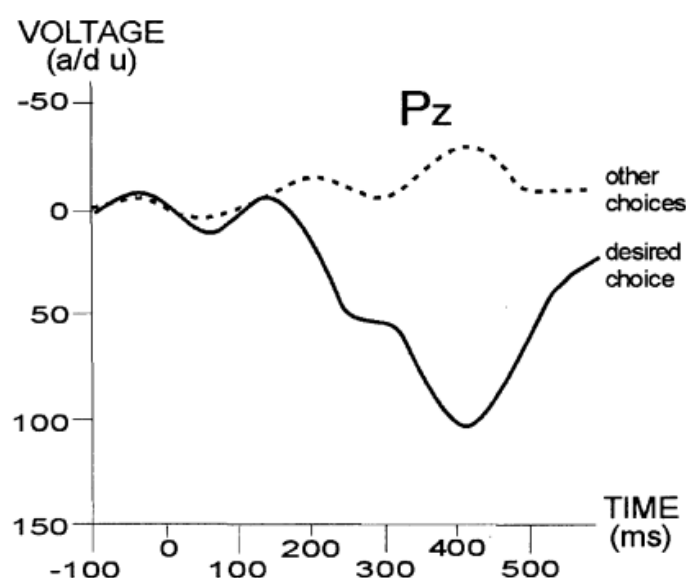
El potencial P300 es la señal ERP más utilizada (*Ilustración 10*). Se trata de una desviación positiva en forma de "pico" que tiene lugar aproximadamente 300 ms después de que ocurra un estímulo poco frecuente, ya sea visual, acústico o táctil [21]. Esta respuesta generalmente se da en la zona parietal y central de la corteza cerebral [5].

Se trata de un potencial que puede ser confundido en presencia de muchos artefactos, por lo cual, se debe hacer una especial énfasis en la prevención y filtrado de los mismos.



Unos de los primeros investigadores que introdujeron la idea de aprovechar los potenciales P300 para diseñar una BCI fueron L. A. Farwell y E. Donchin [22], siendo este último quien la llevó a la realidad en el año 2000 [23].

De forma general, para hacer que funcione una BCI de este tipo, se suele exponer al sujeto a un tren de estímulos de entre los cuales solo unos pocos le pueden llegar a llamar la atención en un momento determinado. Los estímulos son capaces de hacer que se produzca el P300 cuando el usuario muestra ese interés, asociándose cada uno de ellos a una orden concreta de la BCI diseñada [5].



**Ilustración 10: Potencial Relacionado con Evento P300 [18]**

En ciertas investigaciones [14] también se emplea, por ejemplo, el potencial N200, desviación negativa producida 200 ms después de que tenga lugar un estímulo, aproximadamente. El procedimiento para la obtención de los ERP suele ser común en todos los casos, es decir, similar al que se suele llevar con el P300.

#### **4.2.3.2.2. Potencial Visual Evocado**

El potencial visual evocado es una alternativa a los ERP utilizada en ciertas investigaciones. Éste se produce en la corteza visual del cerebro (*Ilustración 8*), causando cambios repentinos en una señal EEG ante un estímulo luminoso [5].

Este tipo de BCIs son de las primeras conocidas, siendo Jackes Vidal quien, entre 1973 y 1977, introdujo la idea de un sistema capaz de determinar el punto sobre el cual un sujeto tiene fija la mirada con objeto de determinar la trayectoria de un cursor [18].



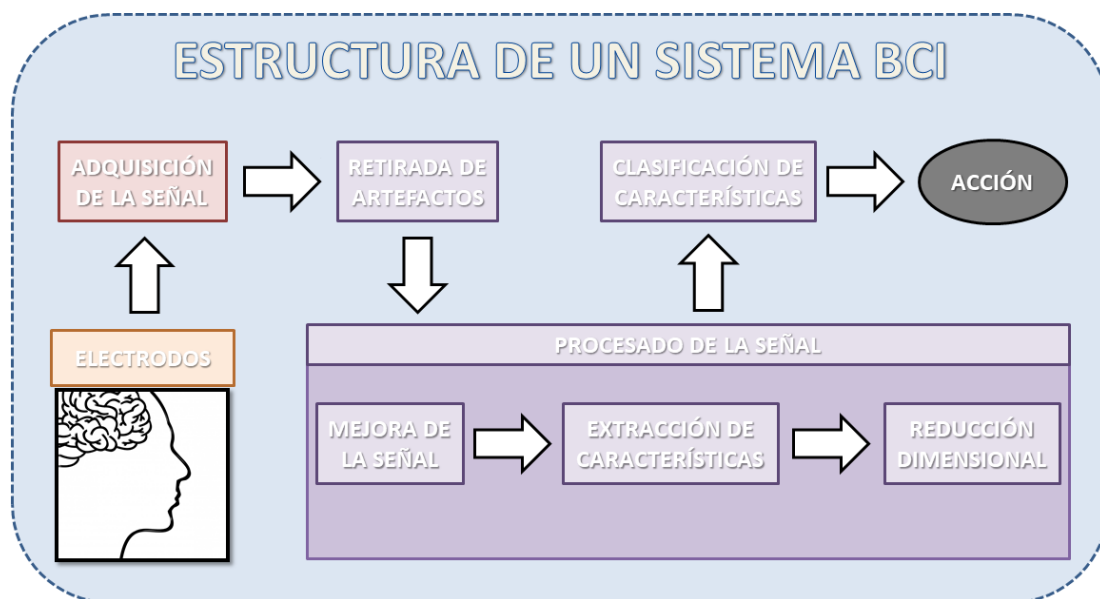
En el caso de que el fenómeno visual que causa la respuesta tenga lugar 6 veces por segundo como mínimo, los potenciales se ven estabilizados, considerándose esta categoría como los Potenciales Evocados Visuales de Estado Estable (SSVEP). Cuando se da esta situación, la amplitud en el espectro aumenta sobre la frecuencia a la que se produce el estímulo, así como en su segundo y tercer armónico [5].

Otra variante es la de los c-VEP, por medio de la cual, los potenciales visuales son modulados a partir de un código aleatorio en lugar de ser frecuencialmente. La idea la introdujo Sutter en una investigación en 1984, llegando a diseñar su propio prototipo de BCI ocho años más tarde. Aun así, hasta 2009 la técnica estuvo en desuso, captando los SSVEP mucho más la atención de los investigadores. Fue G. Bin y su equipo quienes lograron demostrar que el modulado por código era superior al de la frecuencia de los VEP [24].

Un ejemplo de BCI moderna basada en los c-VEP es, por ejemplo, la desarrollada por M. Spuler [24], quien logró utilizar unos estímulos producidos por un código aleatorio para controlar un monitor. Para ello, utilizó la misma codificación binaria para todos los estímulos (de 1.05 segundos de duración), diferenciándolos entre ellos a causa de un lag en la reproducción del código de 0.033 segundos.

#### 4.2.4. Estructura de funcionamiento

Un sistema BCI suele compartir tanto un procedimiento como una estructura común, independientemente de que aplicación se desee implementar y de la actividad cerebral con la cual lo queramos construir.



**Ilustración 11: Diagrama de bloques de un sistema BCI**

En la *Ilustración 11* se puede visualizar un diagrama de bloques que ilustra las diferentes etapas presentes generalmente, desde que la señal es captada por los electrodos hasta que se realiza la acción.

A continuación, se va a desarrollar cada uno de los bloques vistos en la imagen, repasando, entre otras cosas, las técnicas empleadas dentro de diversos estudios.

#### **4.2.4.1. Adquisición de la señal**

El primer paso a dar, una vez captados los impulsos eléctricos, es el diseño de un sistema de adquisición de datos capaz de leer la señal correspondiente a cada uno de los canales de registro.

Para establecer la comunicación entre un dispositivo EEG y el ordenador que se encarga de registrar y procesar los datos, cada instrumento posee sus propios requerimientos. Entre otros, dentro del resumen mostrado previamente en la *Ilustración 5*, se puede ver el método de comunicación utilizado por cada uno de los dispositivos disponibles en el mercado en la actualidad. En él, se indica como, por ejemplo, el *Emotiv* o el *MUSE 2* se conectan con el ordenador gracias al protocolo de *Bluetooth*.

Para lograr la comunicación, es necesario el empleo de un software a través del cual realizar la labor de adquisición de datos, así como el tratamiento posterior de los mismos. El programa puede ser desarrollado desde cero por el usuario dentro de un entorno de programación conocido, o a partir de uno especializado en el tratamiento de señales EEG y diseñado por otra persona o empresa.

En el primer caso, un software muy apropiado para crear un sistema eficaz en la adquisición de datos *online* es *LabVIEW*, plataforma y entorno de programación diseñado por la empresa "*National Instruments*" [25].

Si se opta por la segunda opción, mucho menos laboriosa, es frecuente que las propias compañías de dispositivos EEG ofrezcan su propio kit de desarrollo de software (SDK). Éste, por ejemplo, es el caso de la compañía *Emotiv*, cuya SDK ofrece la posibilidad de adquirir y tratar las señales captadas por su dispositivo para crear una BCI [8]. Por otro lado, hay softwares independientes altamente utilizados en investigaciones y capacitados para cumplir la misma función, como la plataforma *BCI2000* [18] [21].

A su vez, cabe destacar que algunos programas distribuidos para el tratamiento de señales EEG únicamente sirven para la adquisición de datos, necesitando transmitirlos, mediante algún protocolo de comunicación, a otro software capaz de procesarlos. Un ejemplo sería *Mind Monitor* [17], programa citado de forma previa y que es capaz de retransmitir en línea los datos recibidos a través del protocolo *UDP* con cifrado en *OSC*.

A continuación se va a hablar brevemente de los dos protocolos de transmisión, así como del protocolo de cifrado del mensaje, previamente citados.

#### *4.2.4.1.1. Bluetooth de Baja Energía*

El Bluetooth es un protocolo de comunicación de corto alcance muy empleado en el día a día y que tiene como objeto establecer una conexión entre dos dispositivos de forma rápida.

Para ello, los datos son emitidos a través de ondas de radio a baja potencia (2.45 gigahercios aproximadamente), pudiendo prescindir de línea de visión, es decir, pudiendo conectar, por ejemplo, un dispositivo que se encuentre al otro lado de una pared [26]. Estas señales tendrán una amplitud dentro del grado de los milivoltios, evitando así que se pueda causar interferencias con otro tipo de señales pero, sin embargo, limitando así el alcance a 10 metros de distancia.

El protocolo sigue el proceso siguiente [26]:

- Cuando dos dispositivos que funcionan con Bluetooth entran dentro del rango previamente citado, se inicia una comunicación con objeto de saber qué actividad se quiere llevar a cabo.
- Se establece una red, también llamada "*piconet*" que funciona mediante la variación constante y aleatoria de la frecuencia de la señal (1600 veces por segundo) dentro de un ancho de banda designado por la misma.

Por otro lado, tras la implementación del protocolo Bluetooth v4.0, se incorporó un subconjunto que recibió el nombre de Bluetooth de Baja Energía (BLE). Esta forma de comunicación conserva un alcance similar al original reduciendo el consumo de energía, necesitando por lo tanto una fuente de energía grande [27].

A la hora de decidir cuál de las dos formas de comunicación Bluetooth utilizar, el parámetro más importante a valorar es la cantidad de datos a transmitir, ya que de ser elevada, la mejor opción es el método clásico, a expensas de un mayor coste y consumo. Por lo tanto, tal y como su nombre indica, el protocolo resulta muy apto para dispositivos de baja potencia como lo puede ser un MUSE 2.

#### *4.2.4.1.2. Protocolo UDP*

El Protocolo de Datagramas de Usuario (UDP), es un protocolo que tiene como objetivo principal la transmisión de datos de forma rápida gracias a la implementación de una red basada en la IP de los dispositivos a comunicar.

Se trata de una alternativa al Protocolo de Control de Transmisión/Protocolo de Internet (TCP/IP), con la cual se aumenta la velocidad de transmisión, sacrificando la seguridad de que no haya pérdidas de información. Para la comunicación, es esencial la especificación del puerto, comenzando a transmitir los datos sin la necesidad de haber establecido el contacto con un dispositivo de forma previa.

Tal y como se ha comentado, el principal problema del protocolo es la no garantía de que llegue el mensaje al completo, ya que, como el receptor no manda una realimentación, no se puede saber si los datos han llegado de forma correcta. Tampoco se asegura de que el contenido llegue en el orden correcto ni que pueda haber interferido un dispositivo adicional [28].

Cada mensaje UDP consta de una cabecera, la cual se encarga de portar la información que se necesita para establecer una comunicación, así como el conjunto de los datos reales a mandar. La cabecera posee la estructura de la *Tabla 3*, constando de cuatro campos divididos en dos bloques de 32 bits [28].

Bits 0-15		Bits 16-31
Bit 0	Puerto de Origen (Opcional)	Puerto de Destino
Bit 32	Longitud del mensaje	Suma de verificación

**Tabla 3: Estructura de la cabecera en un mensaje UDP [28]**

#### 4.2.4.1.3. Protocolo OSC

El Open Sound Control (OSC) se trata de un protocolo por el cual se redefine el formato de un mensaje y su respectivo contenido con objeto de facilitar la comunicación entre distintos dispositivos, haciéndola más flexible y precisa. Su nombre proviene de su aplicación original, ya que fue pensado para la transmisión óptima de la información que era generada por instrumentos musicales.

Generalmente el transporte del contenido suele ser realizado a través del protocolo UDP, aunque la comunicación también puede realizarse gracias al TCP/IP. Para ello, dentro del protocolo, el “cliente” que transmite los datos realiza un encapsulado de los mismos, dando lugar al formato denominado “argumento” [29], que irá implementado dentro del mensaje.

Tal y como se puede ver en la *Ilustración 12*, este mensaje puede dividirse en tres partes:

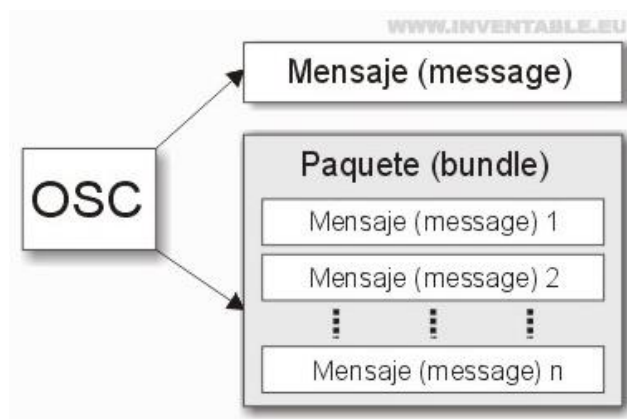
- Una dirección ideada por el usuario y escrita de la misma forma que una URL.

- La abreviatura de los tipos de datos contenidos en los argumentos, como por ejemplo "i" (*entero*), "f" (*coma flotante*) o "s" (*string*).
- Uno o varios argumentos separados por espacios.



**Ilustración 12: Estructura de un mensaje en protocolo OSC [29]**

Un servidor OSC es el encargado de recibir y procesar el mensaje. La unidad de transmisión que capta recibe el nombre de paquete OSC, cuyo contenido puede ser bien un mensaje OSC con su dirección y argumentos, o bien un conjunto de varios mensajes comprimidos en un elemento denominado *paquete* o "*bundle*" [29] (*Ilustración 13*).

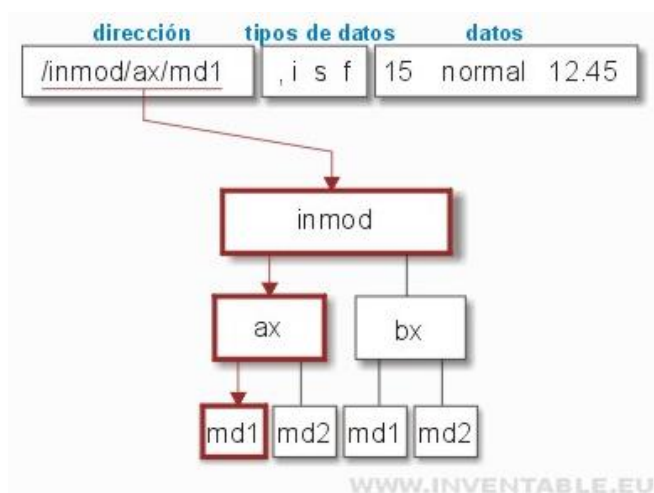


**Ilustración 13: Composición de un paquete OSC [29]**

Los métodos que permiten a un servidor la lectura deseada del mensaje, estando ya predefinidos o habiendo sido diseñados por el usuario, son determinados gracias a la dirección. Presentando una estructura de árbol, la dirección es similar a una URL, realizando la lectura desde la raíz hasta su parte final, tal y como se puede ver en la *Ilustración 14*.

Cuando un cliente envía un mensaje con una dirección concreta, ésta debe coincidir con alguna de las establecidas dentro del servidor, tanto en número de partes como en su contenido. De este modo, el sistema ejecutará el método asociado a dicha dirección, procesando los datos contenidos en el "argumento" una vez son descomprimidos.

Cabe destacar que, pese a que no suelen utilizarse, existen expresiones reguladas capaces de dar dinamismo a los mensajes [30], permitiendo, por ejemplo, aplicar un mismo método a distintas direcciones emitidas por el cliente.



**Ilustración 14: Lectura de una dirección OSC [29]**

#### 4.2.4.2. Retirada de artefactos

Una vez realizada la adquisición de datos, el próximo paso en el diseño de una BCI es la retirada de los artefactos que forman parte de la señal, problema ya comentado con anterioridad.

Esta etapa, puede llevarse de tres maneras diferentes [9]:

- Omitirse, dejando así los artefactos dentro de la señal para, por ejemplo, emplearlos como característica a detectar dentro de la BCI. En el trabajo llevado a cabo por A. N. Belkacem [31], por ejemplo, se logró diseñar un videojuego cuyo manejo es posible gracias a la clasificación de diversos artefactos EOG como el parpadeo.
- Detectar el tramo temporal donde sucede el artefacto para que éste sea eliminado, opción que solo se podría realizar en un procesamiento offline, siendo así poco útil en aplicaciones BCI reales.
- Retirar mediante alguna técnica los componentes de la señal asociados al artefacto, siendo la opción más empleadas en aplicaciones BCI sensibles a este problema. Ejemplos podrían ser las basadas en *Imaginación Motora* y las que emplean *Potenciales Relacionados con Evento (ERP)*.

Centrando el análisis en la última de las maneras, actualmente se ha logrado el diseño de algoritmos de aprendizaje no supervisado capaces de detectar y eliminar una amplia variedad de artefactos. Estos son conocidos como los algoritmos *BBS (Blind Source Separation)* y, entre ellos, se puede considerar tres grupos diferentes [1]:

- PCA (Principal Component Analysis)
- CCA (Canonical Correlation Analysis)
- ICA (Independent Component Analysis)

El llamado en castellano “Análisis de Componentes Principales”, es la técnica más simple de todas. Se basa en la extracción de los PCs, o Componentes Principales de las señales. Esta técnica tiene un problema, al encontrarse en ocasiones los artefactos en correlación con la señal EEG, haciendo que no puedan ser retirados de forma efectiva [1].

Por otro lado, la CCA consiste en la separación de los componentes de fuentes no correlacionadas para detectar una posible correlación lineal entre dos variables [1]. Es frecuentemente utilizada para detectar artefactos musculares.

Finalmente, ICA se basa en la descomposición de las señales EEG en ICs, o Componentes Independientes. Entre éstos, se selecciona los asociados a un tipo de artefacto para que sean retirados, generando una nueva señal con el resto de componentes. Suele dar los resultados más óptimos y por ello es la más usada [10].

A parte de este tipo de algoritmos, de forma independiente, muchos investigadores se han dedicado al diseño de métodos para la detección de artefactos con patrones concretos como, por ejemplo, el producido al parpadear. El proyecto llevado a cabo por M. Agarwal y R. Sivakumar en 2019 [32] consiste en el diseño de un algoritmo no supervisado denominado “Blink”, mediante el cual, detectar de forma automatizada cada vez que un parpadeo es realizado. El algoritmo, que es mostrado en el propio documento, funciona según el estudio a la perfección, fallando únicamente 50 de 2300 parpadeos, de forma aproximada.

#### ***4.2.4.3. Procesado de la señal***

Durante la etapa de procesado, la señal de entrada es transformada para conseguir una serie de valores característicos válidos en la identificación las actividades neuronales a predecir dentro de una BCI.

Este bloque suele resumirse en tres pasos seguidos de forma habitual [19].

##### ***4.2.4.3.1. Mejora de la señal***

Tal y como ya se ha comentado, durante el registro de los impulsos eléctricos, interferencias como el ruido eléctrico son muy habituales. En consecuencia, la señal es distorsionada, generando, por ejemplo, una alta tensión de offset, dificultando la detección de patrones en la misma.

Este paso, que también podría recibir el nombre de *preprocesado*, se basa en la reducción del ruido [19], optimizando la señal al aplicar diferentes técnicas como el uso de filtros digitales. Cabe destacar que, en ocasiones, esta etapa es realizada de forma



## Fundamentos Teóricos y Estado de Arte

previa a la retirada de los artefactos, como, por ejemplo, cuando se emplea la técnica ICA.

Un filtro es un método por el cual se busca alterar la naturaleza de una señal [33], independientemente del tipo que sea. Cuando se habla de un filtro digital, el método por el cual se está procesando la información entrante es a través de una computadora gracias a un algoritmo determinado.

De forma general, tal y como se ha comentado de forma previa, al aplicar un filtro de formato digital, se busca conseguir la eliminación de ciertas frecuencias que resultan perjudiciales a la hora de extraer información de una señal. Los parámetros capaces de regular que rango frecuencial se desea eliminar o, por el contrario, mantener, son las *frecuencias de corte*, puntos a partir de los cuales el filtro intentará atenuar o dejar de atenuar la señal, respectivamente.

En función de los puntos donde se desee ubicar las frecuencias de corte, un filtro puede ser de hasta cuatro tipos distintos:

- Filtro Paso Bajo (LP): Atenúa las frecuencias superiores a una determinada.
- Filtro Paso Alto (HP): Atenúa las frecuencias inferiores a una determinada.
- Filtro Paso Banda (BP): Permite el paso de frecuencias ubicadas en un ancho de banda concreto y conformado por dos frecuencias de corte diferentes (inferior y superior), siendo el resto atenuadas.
- Filtro Elimina Banda o Rechaza Banda (BR): Atenúa las frecuencias dentro de un ancho de banda determinado y conformado, al igual que en el caso anterior, por una frecuencia de corte inferior y otra superior.

Es conveniente destacar que, en aplicaciones reales, entre la completa atenuación de un conjunto de frecuencias (banda de corte) a su paso sin modificaciones (banda pasante), es necesaria una banda de transición [33].

En el presente caso, por ejemplo, las interferencias generadas en señales EEG suelen situarse a frecuencias muy bajas o superiores a los 50 Hz. Por ello, a la hora de aplicar un filtro, éste suele implementarse de paso banda, generalmente una banda pasante de entre 1 a 45 Hz. Además, en algunas situaciones, el acotamiento del espectro de frecuencia debe ser más estrecho, como en las BCIs basadas en Imaginación Motora. En estos casos es necesario observar la banda alpha o beta, cuyas frecuencias de corte han sido mencionadas en el punto 4.1.4.

Por otro lado, en función del procedimiento seguido por el filtro digital que se desee implementar, es conveniente que éstos sean separados en dos clases diferentes, los filtros FIR e IIR.



A grandes rasgos, los filtros de Respuesta Finita al Impulso (FIR) realizan una copia de los datos de entrada, retardándolos uno o varios periodos de muestreo [33]. De forma posterior, éstos son combinados con los nuevos datos contenidos en la señal entrante sin ser retardada.

La expresión matemática que expone lo previamente formulado sería la siguiente [33]:

$$y[n] = a_0 \cdot x[n] + a_1 \cdot x[n-1] + \dots + a_i \cdot x[n-i]$$

- $a_i$  son los coeficientes que conforman el filtro, debiendo ser éstos ajustados, de acuerdo a las frecuencias de corte que se quiera conseguir, de forma previa a su aplicación. El conjunto conforma la respuesta al impulso.
- $x[n]$  es el dato de entrada tomado en un momento concreto.
- $x[n-i]$  es el dato de entrada que se copió  $i$  muestras antes de un momento determinado.
- $i$  depende del orden del filtro, que marca el número de muestras de datos que se va a utilizar para procesar los datos (número de retardos). También es conocido por el nombre de "taps".

Por otro lado, los filtros de Respuesta Infinita al Impulso (IIR) realizan la copia de datos en la salida del filtro, combinándola con las muestras entrantes al realizar una realimentación.

En este caso, la expresión matemática será la siguiente:

$$y[n] = a_0 \cdot x[n] + a_1 \cdot x[n-1] + \dots + a_i \cdot x[n-i] - b_1 \cdot y[n-1] - \dots - b_{ii} \cdot y[n-ii]$$

#### 4.2.4.3.2. Extracción de características

Se trata de una de las partes más importantes dentro de una BCI, teniendo como objeto la búsqueda de ciertos patrones que se puedan relacionar a una actividad neuronal [19].

En función de qué se quiera resaltar, el procesado de la señal será distinto, existiendo actualmente diversas técnicas. Para clasificarlas, se ha utilizado como fuente principal el estudio bibliográfico publicado por B. Medina, J. E. Sierra, y A. B. Ulloa [34].

A modo de resumen, las técnicas empleadas se podrían agrupar en las siguientes:

- Técnicas en el dominio del tiempo.
- Técnicas en el dominio de la frecuencia.
- Técnicas en el dominio del tiempo y frecuencia.
- Técnicas espaciales.

Un resumen del contenido de cada grupo ha sido recopilado dentro de la *Tabla 4* y éste se va a desarrollar a continuación.

Grupo	Técnicas
Dominio del Tiempo	Estadística Instantánea
	Modelos Autorregresivos
Dominio de la Frecuencia	Transformada Rápida de Fourier
	Potencia Espectral
Dominio del Tiempo y Frecuencia	Transformada de Wavelet
	Transformada de Hilbert-Huang
Espaciales	CSP

**Tabla 4: Técnicas para la extracción de características [34]**

Dentro del conjunto de métodos que operan dentro del dominio de tiempo, éstos se basan en detectar irregularidades dentro de la señal dadas en un punto temporal preciso y específico [34]. Este tipo de métodos, por lo tanto, son útiles, por ejemplo, cuando la BCI a desarrollar está basada en *Potenciales Relacionados con Evento*.

Las técnicas normalmente utilizadas para la extracción de características en estos casos pueden ser desde métodos estadísticos, por ejemplo, con el cálculo de la longitud de onda o el cambio de la pendiente en la señal EEG, hasta modelos autorregresivos. Ésta última opción compara una señal original con datos previos gracias a un modelo, evaluando el nivel de correlación [34].

En muchas ocasiones, por ejemplo en el caso de BCIs de *Imaginación Motora*, las características a extraer dependen de cómo se encuentre distribuido el espectro de frecuencia. Por ello, una técnica comúnmente empleada es el paso del dominio del tiempo al de frecuencia gracias a la Transformada Rápida de Fourier (FFT), con la cual se puede conseguir el espectro rápidamente y a tiempo real [34].

Como alternativa a la extracción de características con la FFT, en muchos estudios donde la señal es pasada al dominio de la frecuencia [13], suele resultar más interesante observar la variación de potencia en el espectro. Elevando al cuadrado la transformada de Fourier se puede obtener la Densidad de Potencia Espectral (PSD), siendo esta técnica una de las más utilizadas en aplicaciones BCI [34].

Otra serie de métodos, tal y como se ha podido ver en la *Tabla 4.3*, mezclan el dominio del tiempo y el de la frecuencia. Entre otros, la Transformada de Wavelet es utilizada en ciertas BCIs [35] al permitir analizar una señal EEG en diferentes escalas al mismo tiempo. De esta forma, se permite la evaluación de información a baja frecuencia en intervalos temporales de larga duración y de frecuencias altas en intervalos cortos de forma simultánea. Una alternativa a la Transformada de Wavelet es la de Hilbert-Huang, la cual, según el estudio bibliográfico consultado [34], es muy efectiva frente a señales no estacionarias y no lineales.

Finalmente, hay que destacar el uso técnicas espaciales para la extracción de características, concretamente los Patrones Espaciales Comunes (CSP). Se trata de una serie de filtros que necesitan ser entrenados previamente a su aplicación. El objetivo principal es maximizar la varianza entre dos clases de eventos diferentes proyectando la señal EEG sobre los filtros diseñados [34]. Para ello, se realiza una maximización de los valores asociados a una de las condiciones y una minimización de los de la otra gracias a la proyección de una señal  $x$  a través de una matriz  $W$ .

Matemáticamente, la expresión sería la siguiente [34]:

$$X_{CSP} = W^T \cdot x(t)$$

Se ha demostrado que el empleo de filtros CSP es una de las técnicas más exitosas para la extracción de características en el diseño de un sistema BCI, habiendo sido de gran utilidad para muchos concursantes en competiciones relacionadas con el tema [36]. Generalmente, éstos se utilizan sobre las señales cuando se desea extraer patrones relacionados con la imaginación motora, tras haber utilizado un filtro paso banda para discriminar las frecuencias menos correlacionadas [34].

Sin embargo, a pesar de su popularidad, la técnica presenta algunos inconvenientes al ser altamente sensible al ruido y correr el peligro de ser los filtros sobreentrenados con pocas porciones de datos. Para solucionar problemas de este tipo, muchos investigadores han perfeccionado el algoritmo, dando lugar a variantes como el de Patrones Espaciales Comunes Regularizado (RCSP) [36] [37].

Una última técnica a destacar es la que representa el operador de convolución, ya que, pese a no realizar una extracción de características, es en ocasiones empleado dentro del alisado de la señal tras aplicar, por ejemplo, un filtro CSP.

Una convolución es un operador especial que, tomando una función de entrada  $f(x)$ , la transforma en una nueva de salida gracias a la función  $g(z)$ .

El procedimiento a seguir es siempre el mismo. En primer lugar se debe girar los valores de  $g(z)$  respecto del origen, resultando en la función  $g(-z)$ . De forma posterior, se realiza el desplazamiento de  $g(-z)$  sobre  $f(x)$ , realizando el cálculo de la integral del producto de ambas funciones cada vez que el valor  $x$  varíe en el movimiento de traslación [38].

Matemáticamente, en el caso de una sola dimensión, la operación puede ser representada por medio de la función siguiente, siendo  $*$  el operador que señala que se está realizando una convolución [38]:

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(z) \cdot g(x - z) \cdot dz$$

#### 4.2.4.3.3. Reducción dimensional

En muchas ocasiones, cuando se quiere predecir una actividad determinada, no todos los canales de registro son válidos, ya que, por ejemplo, éste puede situarse en un área que no guarda correlación. En estos casos, el canal empeorará el resultado final cuando se quiera llevar a cabo una clasificación al éstos generar únicamente ruido y no información útil.

Por ello, una vez extraídas las características, se debe reducir la cantidad de datos, dejando solo los importantes y suprimiendo el resto, por ejemplo, mediante la retirada de ciertos canales [19].

#### 4.2.4.4. Clasificación de las características

En esta etapa se realiza la conversión de los datos resultantes del bloque previo en las señales de control lógicas necesarias para que funcione el sistema BCI.

Para ello, se ha de aplicar a las señales previamente procesadas un clasificador que puede ser diseñado mediante dos métodos diferentes:

- Clasificación de las características a partir de algoritmos de aprendizaje supervisado.
- Clasificación de las características a partir de algoritmos de aprendizaje no supervisado.

La principal diferencia entre los dos algoritmos es el método por el cual el clasificador es diseñado. En ambos casos éste ha de ser entrenado pero, sin embargo, cuando el aprendizaje es supervisado, se le debe aportar al algoritmo datos registrados previamente. Para ello, se almacena señales EEG en bruto dentro de bases de datos para que éstas sean procesadas *offline* de la misma manera que se quiere hacer *online* en la BCI. En los registros se debe marcar los periodos en los cuales sucede cada uno de los

eventos a clasificar, permitiendo así al algoritmo generar un modelo capaz de funcionar como clasificador ante señales con patrones similares.

Por otro lado, en el caso de utilizar algoritmos de aprendizaje no supervisado, pese a que el diseño de modelos es mucho más complejo, éstos no precisan de datos previos, calibrando sus parámetros online de forma automática al hacer uso de la BCI.

Dentro del conjunto de algoritmos de aprendizaje supervisado, se puede realizar una división en grupos según la forma de clasificar que tiene el modelo resultante. Ésta puede ser de dos tipos, lineal o no lineal [19].

Dentro de la clasificación lineal, existe un amplio abanico de algoritmos disponibles para poder entrenar un modelo, como, por ejemplo, los métodos clásicos de análisis estadístico [1]. Entre otros, en los estudios previamente consultados, se ha destacado el uso de Regresiones Logísticas (RL) [35], Máquinas de Vector Soporte (SVM) [24] [35] o el Análisis Discriminante Lineal [37].

Por otro lado, el conjunto de técnicas para generar modelos de clasificación no lineal suelen pertenecer al campo de las *Redes Neuronales*. Las más frecuentes son las MLP, que emplean una o dos capas ocultas [1]. A su vez, hay situaciones en las cuales, para diseñar una BCI, se precisa de algoritmos de “*Deep Learning*”, resaltando arquitecturas como la de CNN, GAN, RNN y DNN [1].

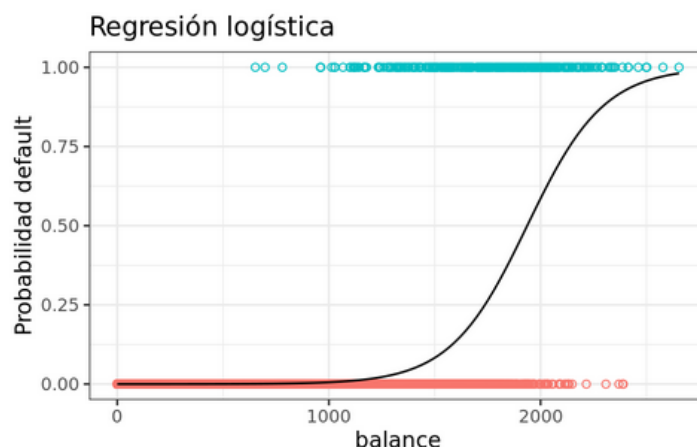
A continuación se va a hablar de forma más detenida acerca de los algoritmos de Regresión Logística y XGboost, los cuales son muy utilizados en el campo del “*Machine Learning*” y a través de los cuales se va a realizar modelos en el presente proyecto.

#### 4.2.4.4.1. *Regresión logística*

La Regresión Logística es una técnica, situada en el conjunto de Modelos Lineales Generalizados (GLM), que permite establecer la probabilidad de que se esté dando una variable binaria cualitativa a partir de ciertas variables cuantitativas. Por consiguiente, su aplicación más habitual es la construcción de un modelo capaz de clasificar dos categorías (regresión binaria), calculando para ello el logaritmo de la probabilidad de que suceda cada una de ellas en un momento determinado.

Hay dos tipos diferentes:

- Regresión logística simple: Una sola variable de entrada (*Ilustración 15*).
- Regresión logística múltiple: Varias variables de entrada.



**Ilustración 15: Regresión Logística Simple y Binaria [39]**

El método está basado en el algoritmo de Regresión Lineal, a través del cual es posible la clasificación de variables cuantitativas ajustando los parámetros  $\beta$  para conformar la ecuación de una recta que permita estimar los resultados con el mínimo error:

$$\text{Función lineal} = \beta_0 + \beta_1 \cdot X$$

En el caso de querer predecir un par de variables cualitativas, lo más sencillo sería transformar los resultados de la Regresión Lineal para que únicamente se pueda obtener valores cercanos a "0" o "1". El algoritmo de Regresión Logística es uno de los más idóneos en situaciones de este tipo [40], ya que, aplicando una función sigmoide, al procesar resultados  $x$  grandes, la salida es cercana a "0" y, por el contrario, con números bajos, resulta en "1" [39].

$$\text{Función sigmoide (logística)} = \frac{1}{1 + e^{-x}}$$

Partiendo de una tarea de regresión simple y binaria, al sustituir el parámetro  $x$  por la ecuación que define la recta, se obtiene la siguiente función, siendo  $p$  la probabilidad de obtener cada una de las variables de salida "Y" y  $X$  el dato de entrada:

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot X)}} = \frac{e^{(\beta_0 + \beta_1 \cdot X)}}{1 + e^{(\beta_0 + \beta_1 \cdot X)}}$$

Es esencial el ajuste de los parámetros  $\beta$  para el buen funcionamiento del modelo. Los métodos de regresión lineal son los más sencillos de aplicar [39] y, para lograrlo, se debe dejar a un lado de la ecuación la parte que corresponde a la composición de la recta, resultando así en una ecuación logarítmica:

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \cdot X$$

Para ajustar los parámetros, hay que introducir el concepto de ODD, o razón de probabilidad [39]. Éste sería la relación entre la probabilidad  $p$  de que la categoría pueda ser "1" y la probabilidad "q" de que sea la contraria. Por lo tanto:

$$q = 1 - p$$

$$ODD = \frac{p}{q} = \frac{p}{1 - p}$$

Finalmente, calculando el logaritmo neperiano del ODD se obtiene un resultado dentro del rango  $[-\infty, \infty]$  [38], pudiéndose llegar a la siguiente conclusión:

$$\ln\left(\frac{p}{q}\right) = \ln(ODD) = \beta_0 + \beta_1 \cdot X$$

De este modo, introduciendo un conjunto de resultados y sus correspondientes variables de entrada, se puede ajustar  $\beta_0$  y  $\beta_1$  de forma lineal una vez hecho el cálculo logarítmico aplicado a los ODDs. Para ello, los métodos más comúnmente utilizados son el de Newton y el del descenso de gradiente [38]. El modelo resultante debe de ser el más simple que, a su vez, logre el mínimo grado de error (principio de parsimonia) [40].

Una vez optimizada la ecuación, ya se podría empezar a aplicar el modelo a partir de nuevas variables  $X$ , utilizando la fórmula con la cual obtener  $p$ .

En el caso de realizar una Regresión Logística Múltiple, la mayor diferencia sería el aumento de parámetros  $\beta$  a ajustar, empleando por lo demás principios similares:

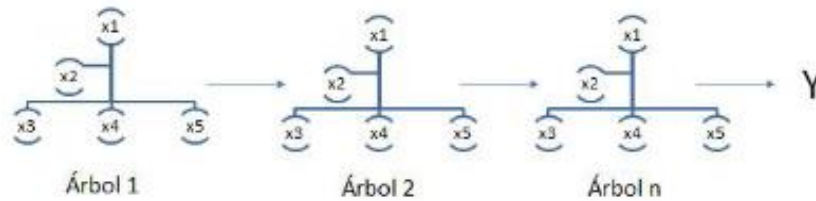
$$\ln\left(\frac{p}{q}\right) = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \dots + \beta_i \cdot X_i = \beta_0 + \sum_{i=1}^n \beta_i \cdot X_i$$

$$p = \frac{e^{(\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \dots + \beta_i \cdot X_i)}}{1 + e^{(\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \dots + \beta_i \cdot X_i)}} = \frac{e^{(\beta_0 + \sum_{i=1}^n \beta_i \cdot X_i)}}{1 + e^{(\beta_0 + \sum_{i=1}^n \beta_i \cdot X_i)}}$$

#### 4.2.4.4.2. XGboost

El algoritmo de *Extreme Gradient Boosting*, más conocido como *XGBoost*, es una de las técnicas que emplean árboles de decisión más utilizadas actualmente en labores de aprendizaje supervisado.

Está basada en el principio de *boosting*, es decir, en el diseño de una sucesión de modelos predictivos, partiendo de uno de bajas capacidades y, con los resultados obtenidos, ajustando ciertos parámetros y generando versiones nuevas y mejores (Ilustración 16). Generalmente, esta optimización es lograda gracias al algoritmo de *Descenso del Gradiente* [41].



**Ilustración 16: Algoritmo de XGBoost [42]**

La capacidad predictiva de los modelos nuevos es comparada con la del previo, descartándolo en caso de obtener peores resultados. De este modo, el proceso continúa hasta que no se aprecie grandes cambios entre los valores obtenidos en los últimos dos modelos o cuando se llegue al número máximo de iteraciones [41].

Matemáticamente, el algoritmo se podría definir con la siguiente expresión [43]:

$$\hat{y}_i^{(t)} = \sum_{k=1}^t f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)$$

- $t$  es el número de árbol de decisión.
- $f(x_i)$  es la función utilizada para realizar la predicción.
- $\hat{y}_i^{(t)}$  es el resultado de la predicción del modelo.

Una vez se obtenida la predicción, ésta debe de evaluarse para ver cómo es de bueno el modelo. Por ello, se hace uso de la denominada función objetivo [43], cuyo resultado debe de ser menor cada vez que se evalúe un nuevo modelo:

$$obj^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)})$$

- $l$  es la función a través de la cual comparar el resultado del modelo.
- $y_i$  es el valor real de la predicción que espera lograr.

Generalmente,  $l$  está definida para realizar el cálculo del Error Cuadrático Medio, calculando así la diferencia entre el valor real y el predicho [44]:

$$l(y_i, \hat{y}_i^{(t)}) = \sum_i (y_i - \hat{y}_i^{(t)})^2$$

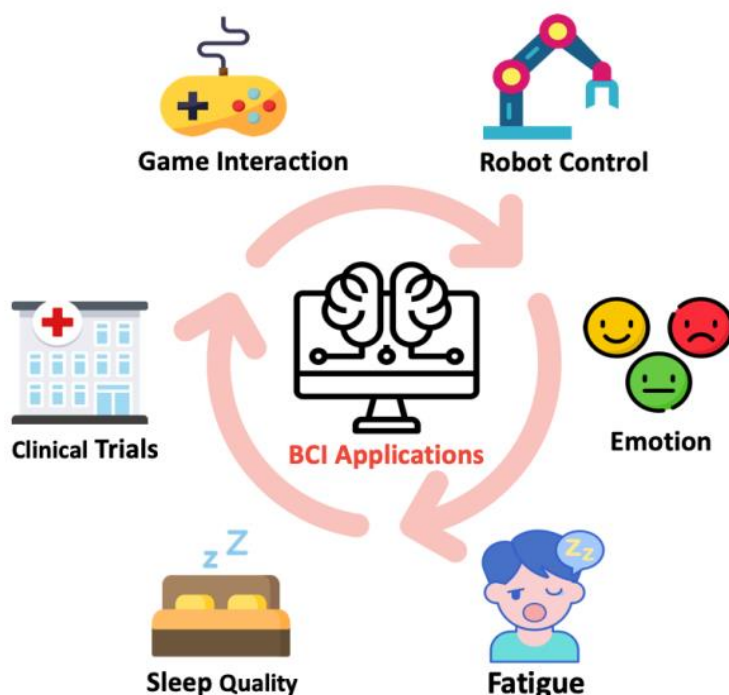
#### 4.2.5. Aplicaciones

Según el estudio bibliográfico de X. Gu y su equipo [1], muchas investigaciones recientes dentro del diseño de Interfaces Cerebro-Computadora han girado en torno a la neurocomputación. En ellas, principalmente se busca innovar en las técnicas para la



búsqueda de patrones en las señales EEG, así como técnicas para entrenar nuevos tipos de clasificadores.

También cabe destacar que, en la actualidad, gracias al estudio de los sistemas BCI, éstas han contribuido en el progreso de diferentes campos de la investigación [1], como lo puede ser el de las emociones o el estudio de la calidad del sueño. Entre otros, estas categorías pueden verse representadas en la *Ilustración 17*.



***Ilustración 17: Campos de investigación donde han contribuido las BCI [1]***

Por otro lado, el estudio de X. Gu y su equipo [1] también establece una separación en torno a los campos de aplicación que han tenido los sistemas BCI desarrollados de forma reciente. A continuación se va a proceder a hablar acerca de los mismos.

#### ***4.2.5.1. Interfaces Humano-Computadora (HCI)***

Dentro de este ámbito, el diseño de sistemas BCI busca evitar la utilización de elementos manuales (ratón y teclado, por ejemplo) para la comunicación entre persona y computadora.

Históricamente la idea de controlar un ordenador con el pensamiento ha sido muy explotada, pudiéndose encontrar un gran número de estudios que lo han intentado a partir de dispositivos de investigación.

Algunos trabajos han mostrado BCIs capaces de controlar punteros a partir de imaginación motora gracias a los ERD tras extraer el espectro de potencia (PSD) [13].

Otros han conseguido dotar a un sujeto de la capacidad de moverse por aplicaciones Windows, así como escribir en ellas, gracias a los c-VEP [24] o, por ejemplo, navegar por internet utilizando los potenciales P300 [45].

Un ejemplo citado en el estudio bibliográfico consultado [1] muy interesante es la elaboración de un sistema orientado a ayudar en el Diseño Asistido por Ordenador (CAD) [46]. En él, es diseñada una BCI que, gracias a, por ejemplo, artefactos EOG o EMG, es capaz de controlar un programa de modelado en 3D a través de un dispositivo EEG de Emotiv.

#### ***4.2.5.2. Entretenimiento***

Aprovechando el auge de los dispositivos EEG portátiles, dentro del mundo del entretenimiento, el de los videojuegos concretamente, se está comenzando a plantear innovaciones orientadas al diseño de sistemas BCI.

En este ámbito, tal y como se ha citado previamente, es muy común aprovechar los artefactos como patrones a detectar, generando de este modo los comandos por los cuales controlar el videojuego. En estos casos, la experiencia puede ser mejorada a través de la implementación de sensores extra con objeto captar eventos adicionales como las expresiones faciales [1]. Un ejemplo de este tipo de BCI podría ser el diseñado por A. N. Belkacem y su equipo [18] o el de T. McMahan, I. Parberry, y T. D. Parsons [47].

Por otro lado, otro enfoque que se le puede dar a un sistema BCI dentro del área del entretenimiento es por medio de la predicción de estados mentales. En la última década se ha llevado a cabo exitosamente varias investigaciones buscando la innovación en cuanto a los métodos de extracción y clasificación de características [35] [48].

A causa de los últimos avances, en estudios como el dirigido por J. Teo y J. T. Chia [49], se ha llegado a plantear la extracción de las respuestas ante diferentes estados mentales para servir de realimentación ante un videojuego. De este modo, fácilmente se indicaría el estado del usuario, mostrándole al sistema cómo actuar, mejorando así la experiencia. Finalmente, el proyecto logró, a través de redes neuronales de aprendizaje profundo, la clasificación exitosa del estado de excitación causado por los estímulos generados en un entorno de realidad virtual.

#### ***4.2.5.3. Bienestar de las personas***

Dentro de este campo, muchas de las investigaciones están motivadas en poder mejorar la calidad de vida de personas que presenten algún tipo de discapacidad.

En este contexto, según lo comentado en el artículo del CSIC citado previamente [5], las aplicaciones más habituales que uno se puede encontrar están orientadas al ámbito de la comunicación. Es decir, se trata de BCIs capaces de ayudar a personas con problemas en el habla, permitiendo que éstas puedan interactuar con el entorno a partir de la extracción de los P300, SCP o basando la interface en Imaginación Motora [5].

Muchas aplicaciones en este ámbito en concreto también podrían ser incluidas dentro de la categoría de HCI, al tener implementada la capacidad de navegar por Internet e, incluso, participar en redes sociales, publicando mensajes en *Twitter*, por ejemplo [5].

Otro tipo de aplicaciones estudiadas en la actualidad son las que pretenden aportar facilidades a personas con problemas de movilidad reducida. En estos casos, se puede encontrar dos tipos distintos de BCI:

- Control de un sistema que permita el libre desplazamiento del usuario a través de un entorno como, por ejemplo, una silla de ruedas
- Control de un sistema que permita a un usuario suplir la carencia de alguna extremidad como, por ejemplo, a través de un sistema robotizado.

En el primero de los casos, por ejemplo, el control de un sistema de este tipo puede estar basado en P300 o en Imágenes Motoras [5]. Un ejemplo de ello podría ser el trabajo realizado en 2009 por I. Iturrate, J. M. Antelis, A. Kubler, y J. Minguez [21], donde se diseñó una BCI capaz de controlar la trayectoria de una silla mediante la clasificación de potenciales P300 a través del algoritmo SWLDA.

Por otro lado, los estudios que tratan el diseño de sistemas robotizados pueden abarcar tanto el manejo de robots móviles [8] como el de brazos robóticos para, por ejemplo, la manipulación de objetos [50].

Por último, cabe destacar otras variantes de sistemas BCI orientados a mejorar el bienestar de las personas pero que no pueden categorizarse en las clases expuestas de forma previa. Entre otros, cabe destacar el planteamiento de un estudio para crear un sistema de control domótico [5] o, por ejemplo, la investigación de una BCI capaz de controlar un dron por medio de un dispositivo del grado de consumo [51].

## 4.3. MÉTRICAS PARA LA EVALUACIÓN DE UN MODELO

### 4.3.1. *Introducción*

Tal y como ya se ha comentado en el apartado 4.2.4, cuando se le aplica a un conjunto de datos un algoritmo de aprendizaje supervisado, se genera un modelo con el cual poder realizar las predicciones necesarias dentro de una BCI. Para que el sistema funcione correctamente, por lo tanto, se debe conseguir el diseño de un clasificador óptimo, debiéndose evaluar a través de ciertas métricas los resultados obtenidos al aplicarlo sobre un nuevo conjunto de datos de testeo.

En el presente proyecto, la metodología a seguir se ha basado en la empleada en el estudio de R. H. Gatti, L. G. Escher, y L. C. Carrere [37], añadiendo a su vez medidores típicos utilizados en el campo de la inteligencia artificial. Por ello, es conveniente destacar que, a partir de este momento, se va a hablar acerca métricas consideradas para evaluar un modelo binario, entrenado con datos de señales EEG ya procesadas y a partir de eventos basados en Imaginación Motora.

El procedimiento mostrado en el estudio [37] diferencia dos tipos de métricas a la hora de poder valorar la capacidad predictiva de un modelo, una encargada de puntuar la capacidad de Imaginación Motora del sujeto y otra para evaluar de los resultados de testeo (coeficiente Kappa de Cohen). Sin embargo, tal y como ya se ha dicho, a su vez se ha considerado importante destacar una serie de indicadores adicionales dentro del último apartado con objeto de aumentar la calidad de la información obtenida.

### 4.3.2. *Capacidad de Imaginación Motora del sujeto*

Hay estudios que demuestran que no todos los sujetos tienen la misma capacidad de imaginar movimientos sin realizarlos en la realidad. Para poder comprobar la habilidad que posee cada uno de los sujetos a los cuales se les ha realizado un registro de datos, tal y como se ha indicado dentro de la fuente consultada [37], se ha de hacer un cuestionario capaz de puntuar con rigor científico.

Durante la última mitad del siglo XX, se configuraron varios test capaces de cuantificar estas capacidades. En su tiempo, el más famoso fue el denominado “*Movement Imagery Questionnaire*” (MIQ), que, diseñado en los años 80, se consideró como el más fiable y, por lo tanto, utilizado [52].

En 1997, el cuestionario fue sometido a modificaciones a causa de las siguientes razones [52]:

- Reducir el número de pruebas (*ítems*).
- Modificar las puntuaciones.
- Alterar ligeramente la forma de realizar alguna de las pruebas.

Como resultado dio lugar el MIQ-R, que rápidamente sustituyó al cuestionario original, siendo traducido al castellano por A. Campos y M. A. González en el año 2010 [52]. Dentro del mismo, cada una de las pruebas o ítems consta de los siguientes pasos:

- Descripción de la posición inicial para realizar un movimiento y realización por parte del sujeto.
- Descripción del movimiento y realización del mismo por parte del sujeto.
- Vuelta a la posición inicial y ejecución del mismo movimiento pero en este caso únicamente a través de imaginación motora.
- Puntuación de la facilidad del sujeto a la hora de realizar el paso previo.

Los ítems son divididos en dos subescalas encargadas de valorar aspectos distintos de la capacidad del sujeto a la hora de imaginarse un movimiento [52]:

- Capacidad de formar una imagen mental del movimiento (Imagen Visual).
- Capacidad de sentir un movimiento sin llegar a hacerlo realmente (Imagen Cinestésica).

Por otro lado, las valoraciones son realizadas en la escala del 1 al 7, siendo, los valores bajos, indicadores de poca habilidad en actividades de imaginación motora, y por consiguiente, los elevados muestran lo contrario.

### *4.3.3. Evaluación de los resultados del clasificador*

#### *4.3.3.1. Matriz de Confusión*

El conjunto de datos básico que se necesita para evaluar la capacidad de predicción de los modelos construidos son los contenidos en la Matriz de Confusión (*Tabla 5*).

Es importante tener en cuenta que, cuando se valoran los resultados de un clasificador binario, un evento será identificado numéricamente por la etiqueta "0" (*negativo*) y el otro como "1" (*positivo*) [53]. Por otro lado, cuando la predicción sea cierta comúnmente se refiere a un valor *verdadero*, y si el modelo *falla*, falso.

MATRIZ DE CONFUSIÓN			
	Característica Predicha: 0	Característica Predicha: 1	Total
Característica Real: 0	$C_{0,0}$	$C_{0,1}$	$C_{0,0} + C_{0,1} = C_{0,:}$
Característica Real: 1	$C_{1,0}$	$C_{1,1}$	$C_{1,0} + C_{1,1} = C_{1,:}$
Total	$C_{0,0} + C_{1,0} = C_{:,0}$	$C_{0,1} + C_{1,1} = C_{:,1}$	$C_{0,0} + C_{0,1} + C_{1,0} + C_{1,1} = N$

**Tabla 5: Parámetros de una matriz de confusión**

Los parámetros contenidos en la tabla son los siguientes:

- $C_{0,0}$ : Coeficiente de la matriz que indica el conjunto de eventos 0 que ha acertado el modelo en la clasificación (también llamado "Verdadero Negativo").
- $C_{0,1}$ : Coeficiente de la matriz que indica el conjunto de eventos 0 que el modelo ha confundido como 1 (también llamado "Falso Positivo").
- $C_{1,1}$ : Coeficiente de la matriz que indica el conjunto de eventos 1 que ha acertado el modelo en la clasificación (también llamado "Verdadero Positivo").
- $C_{1,0}$ : Coeficiente de la matriz que indica el conjunto de eventos 1 que el modelo ha confundido como 0 (también llamado "Falso Negativo").

Con los valores citados, se puede evaluar el balanceo de los datos, pudiendo intuir la calidad de un clasificador entrenado.

A su vez, a partir de la matriz de confusión, se puede calcular el conjunto de parámetros que permiten la valoración numérica del modelo.

#### 4.3.3.2. Exactitud

Esta medida, que se encarga de representar numéricamente el porcentaje de aciertos, tanto positivos como negativos, es la medida más comúnmente utilizada a la hora de valorar los resultados de un modelo [53]. Generalmente se suele referir a ella por su denominación en inglés, es decir, "accuracy".

En el estudio [37], la forma de llamar a este parámetro ha sido  $p$ , y la forma de expresarlo matemáticamente es la siguiente:

$$p = \frac{\sum C_{i=j}}{N} = \frac{C_{0,0} + C_{1,1}}{C_{0,0} + C_{1,0} + C_{0,1} + C_{1,1}}$$

- $C_{i=j}$  son el conjunto de coeficientes contenidos en la matriz de confusión en los cuales el modelo ha acertado, es decir, verdaderos ( $C_{0,0}$  y  $C_{1,1}$ ).
- $N$  es la suma de los eventos totales clasificados.

Sin embargo, pese a su frecuencia a la hora de utilizarse, este parámetro es el que peor calidad de información aporta, ya que puede revelar un valor positivo al predecir de forma correcta una de las clases pero estar fallando en la otra sin poder percibirlo. Es por ello que se necesita calcular una serie de medidores adicionales.

#### 4.3.3.3. Precisión

Una forma sencilla de solucionar el problema visto en el apartado anterior es a través del cálculo de la precisión de acierto de cada una de las categorías. Para ello, se calcula el porcentaje de verdaderos positivos y verdaderos negativos por el total de positivos y negativos predichos, extrayendo así la información sobre los aciertos producidos a la hora de clasificar una característica en concreto.

Matemáticamente, la manera a través de la cual se puede realizar el cálculo de la precisión para ambas categorías se puede ver representado a partir de las siguientes dos fórmulas:

$$precisión_0 = \frac{C_{0,0}}{C_{0,0} + C_{0,1}}$$

$$precisión_1 = \frac{C_{1,1}}{C_{1,0} + C_{1,1}}$$

#### 4.3.3.4. Exhaustividad

Por otro lado, otra medida útil a la hora de analizar cada una de las categorías a predecir por separado, es la exhaustividad, más conocida por el nombre de "recall" [53].

Se trata de comparar el número de aciertos del modelo sobre una clase frente al número total de veces que se tendría que haber predicho. Es decir, en este caso, no se trata de comprobar numéricamente el número de veces que se ha acertado a la hora de hacer una predicción, sino que se cuantifica las ocasiones en las cuales el modelo ha acertado en comparación con las que lo debería haber hecho.

A la hora de representarlo de forma matemática, las ecuaciones empleadas son las siguientes:

$$exhaustividad_o = \frac{C_{0,0}}{C_{0,0} + C_{1,0}}$$

$$exhaustividad_1 = \frac{C_{1,1}}{C_{0,1} + C_{1,1}}$$

#### 4.3.3.5. Valor F1

En caso de que se desee valorar un modelo valorando por igual tanto la precisión como la exhaustividad, es frecuente la utilización del valor F1, o también llamado "F1-score".

Se trata de la media armónica entre los dos parámetros ya mencionados, pudiendo expresarse matemáticamente de la forma siguiente [53]:

$$F1 = 2 \cdot \frac{precisión \cdot exhaustividad}{precisión + exhaustividad}$$

#### 4.3.3.6. Coeficiente Kappa de Cohen

El Coeficiente Kappa de Cohen (k) es la medida utilizada en el estudio de R. H. Gatti, L. G. Escher, y L. C. Carrere [37] para valorar la calidad del modelo diseñado. Éste permite la verificación numérica de la probabilidad de que el clasificador realice cada una de sus predicciones al azar o porque realmente se ha encontrado un patrón.

Para ello inicialmente se debe haber calculado la exactitud p, a partir de la cual poder aplicar la ecuación siguiente:

$$k = \frac{p - p_0}{1 - p_0}$$

- p es el porcentaje de exactitud del modelo.
- p<sub>0</sub> es la probabilidad de aciertos si la elección se realiza al azar.

Tal y como se puede comprobar, de forma previa al cálculo, es necesario obtener p<sub>0</sub>, utilizándose la fórmula mostrada a continuación:

$$p_0 = \frac{\sum C_{i,:} \cdot C_{:,j}}{N^2}$$

$$p_0 = \frac{C_{:,0} \cdot C_{0,:} + C_{:,1} \cdot C_{1,:}}{(C_{0,0} + C_{1,0} + C_{0,1} + C_{1,1})^2}$$

$$p_0 = \frac{(C_{0,0} + C_{1,0}) \cdot (C_{0,0} + C_{0,1}) + (C_{0,1} + C_{1,1}) \cdot (C_{1,0} + C_{1,1})}{(C_{0,0} + C_{1,0} + C_{0,1} + C_{1,1})^2}$$



- $C_{i\cdot}$  es la suma de los coeficientes que componen la fila  $i$  dentro de la Matriz de Confusión.
- $C_{\cdot j}$  es la suma de los coeficientes que componen la columna  $j$  dentro de la Matriz de Confusión.

Una vez obtenido  $k$ , se debe seguir un criterio determinado de cara a valorar el resultado. Para ello, una posibilidad es la propuesta por Landis y Koch [54], los cuales establecieron unos límites con los que poder estimar el grado de acuerdo de los dos observadores del modelo (etiqueta real y etiqueta escogida por el clasificador). Esta información se ha recogido en la *Tabla 6*.

Coeficiente $k$	Grado de Acuerdo Estimado
$k < 0.0$	Nulo
$0.0 < k \leq 0.2$	Insignificante
$0.2 < k \leq 0.4$	Bajo
$0.4 < k \leq 0.6$	Moderado
$0.6 < k \leq 0.8$	Bueno
$0.8 < k \leq 1.0$	Muy Bueno

**Tabla 6: Grado de acuerdo estimado en función del coeficiente  $k$**

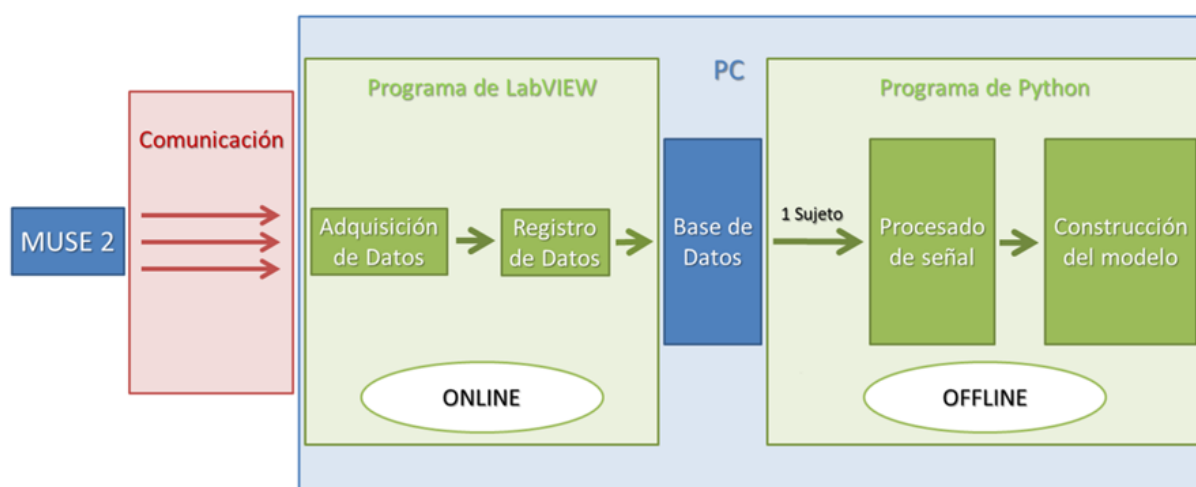
## 5. MÉTODO OPERATIVO

### 5.1. INTRODUCCIÓN

Dentro del Trabajo de Fin de Grado se pretende, por medio de un dispositivo EEG del grado de consumo, realizar la adquisición de las señales correspondientes a cada uno de los canales de registro para su etiquetado y almacenamiento en diferentes *databases*. Posteriormente, se quiere procesar el conjunto de los datos captados para poder entrenar una serie de algoritmos de aprendizaje supervisado, resultando en el diseño de modelos para clasificación lineal y binaria de un par de pensamientos de imaginación motora.

El método operativo que se va a llevar a cabo ha sido representado en la *Ilustración 18*, pudiendo ser resumido en tres grandes bloques:

- Comunicación ente el dispositivo Muse y el ordenador, de cara a poder transmitir las señales en bruto captadas por los canales hasta un ordenador, lugar donde éstas van a poder ser manipuladas gracias a la programación.
- Programa de LabVIEW encargado de adquirir las señales a tiempo real, así como de permitir que éstas puedan etiquetarse y registrarse, generando archivos CSV que posteriormente puedan procesarse offline.
- Programa de Python encargado del procesamiento de las bases de datos obtenidas. Realmente esta parte se podría dividir en dos, separando el tratamiento aplicado a las señales en bruto iniciales de la utilización de algoritmos de inteligencia artificial para generar el modelo óptimo, una vez extraídas las características.



**Ilustración 18: Diagrama del método operativo seguido**

## 5.2. COMUNICACIÓN

Para establecer una comunicación entre dispositivos mediante Muse 2, éste cuenta con conexión por medio de Bluetooth de Baja Energía (BLE). Tal y como se ha comentado en el *punto 4.2.4.1.1* este protocolo permite una buena transmisión de los datos dentro de la corta distancia, evitando que se dé un consumo elevado de energía. Sin embargo, la comunicación directa entre este dispositivo y un ordenador es difícil, ya que se presentan incompatibilidades que dificultan la transmisión.

Una posible solución hubiera sido la compra de un adaptador que, debido a la posibilidad de que aun así tampoco fuera a funcionar, finalmente se rechazó. En su lugar, se ha optado por emplear un móvil como puente para que, de este modo, MUSE 2 se conecte a un dispositivo con mayor compatibilidad y éste se encargue de mandar los datos al ordenador.

La aplicación gracias a la cual se ha podido realizar tal labor ha sido "*Mind Monitor*" [17]. Este programa, creado por un autor independiente a la empresa de MUSE, puede ser adquirido a través de la tienda habilitada dentro del dispositivo móvil.

La comunicación se establece directamente con MUSE 2 por BLE, permitiéndonos la visualización online de los datos EEG leídos por los electrodos, los cuales pueden ser registrados en archivos CSV gracias a una opción que posee la interface.

Sin embargo, para seguir adelante con los objetivos planteados en el TFG, se ha optado por otra de las posibilidades que ofrece el programa, que es la transmisión de los datos online por medio de la comunicación UDP (*punto 4.2.4.1.2*). Previamente, el contenido del mensaje es codificado siguiendo el protocolo OSC (*punto 4.2.4.1.3*).

Para comprobar que el ordenador recibe datos procedentes del dispositivo móvil, se ha usado el programa "*Wireshark*" [55], el cual permite comprobar los movimientos que suceden dentro de la red WIFI por donde se transmite la información.

## 5.3. LABVIEW

### 5.3.1. Adquisición de datos

El programa elegido para la adquisición y registro de los datos online es, tal y como se ha dicho previamente, LabVIEW [25], un software que permite el diseño de programas complejos a través de la comunicación entre diferentes bloques.

## Método Operativo

En primer lugar, el objetivo que se tiene es la adquisición de los datos que son transmitidos por UDP a un puerto concreto. Para ello, se ha realizado un programa que funcione como servidor capaz de recibir los mensajes procedentes del dispositivo móvil.

Una vez el conjunto de mensajes puede ser leído gracias al programa, éste último es desarrollado para poder decodificar el formato OSC. Para ello, se ha descargado la biblioteca "NI OSC" [56], facilitando en gran medida la lectura de los datos gracias a las funciones prediseñadas que implementa una vez se ha instalado.

Las direcciones codificadas dentro del mensaje, que permiten distinguir el tipo de dato que se está recibiendo, se encuentran publicadas en la web de Mind Monitor [17], así como la frecuencia de muestreo a través de la cual se transmite cada uno de ellos. Esta información se ha resumido en la *Tabla 7*, la cual se puede ver a continuación.

Tema de Datos	Dirección OSC	Valores	Datos	Unidades	Frecuencia
<b>EEG en Bruto</b>	/muse/eeg	Flotante	TP9, AF7, AF8, TP10	Microvoltios	220 256 Hz
<b>Calidad de conexión</b>	/muse/elements/horseshoe	Flotante	TP9, AF7, AF8, TP10	1 = Buena 2 = Media 4 = Mala	10 Hz
<b>Giroscopio</b>	/muse/gyro	Flotante	X, Y, Z	Grados/seg	10 Hz
<b>Acelerómetro</b>	/muse/acc	Flotante	X, Y, Z	g {-2,2}	10 Hz

**Tabla 7: Información de interés sobre los datos transmitidos por Mind Monitor**

Una vez se ha logrado decodificar los mensajes en formato OSC, se ha pasado a diseñar una interface que permita la visualización online del conjunto de datos recibidos. Para ello, se ha implementado un total de cuatro gráficas encargadas de imprimir las señales EEG en bruto procedentes de los canales de registro. A su vez, por medio de una barra de progreso, se ha habilitado un indicador para ver la calidad de contacto entre los electrodos y la cabeza del sujeto, así como un par de gráficas para mostrar los datos del acelerómetro y el giroscopio.

### 5.3.2. Registro de datos

Una vez realizada correctamente la adquisición de datos, el próximo paso ha sido el registro de la señal en bruto con objeto de construir una base de datos.

En primer lugar, se ha procedido a modificar nuevamente el programa de LabVIEW con objeto de implementar la posibilidad de grabar los datos procedentes del MUSE 2 durante un periodo de tiempo concreto. Como resultado, una vez se decida detener el

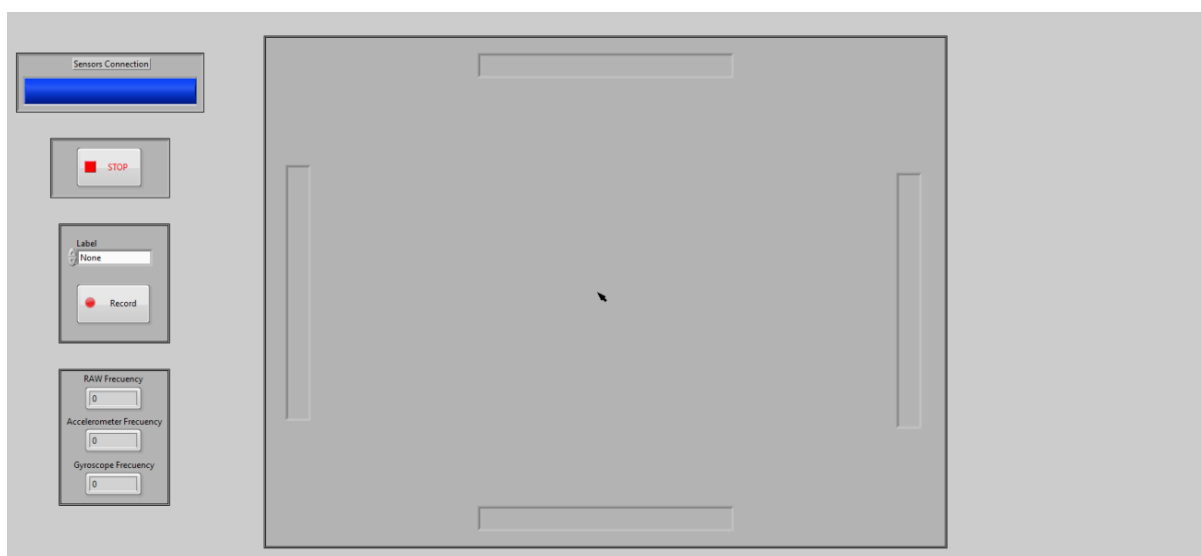
programa, se guardará un archivo en formato CSV con la información registrada, que a partir de ahora se va a denominar "*dataset*".

Estos datos han de ser etiquetados para que, cuando sean procesados offline, se puedan asociar a un evento determinado. Por esa razón, se ha diseñado el programa de forma que, junto con las señales EEG, en el *database* se guarde una variable "*string*" que pueda indicar el evento que está sucediendo en un momento determinado. En este caso, estas etiquetas serán tres en total, mostradas a continuación en la *Tabla 8*.

Etiqueta	Tipo de datos etiquetados
<b>None</b>	Conjunto de datos no válidos para la construcción de un modelo. Se trata de información que ha sido registrada en momentos de descanso, es decir, sin producirse ningún tipo de evento.
<b>Right</b>	Datos registrados mientras se está haciendo un ejercicio de imaginación motora en el que un sujeto imagina desplazar la mano diestra hacia la derecha para mover un ratón.
<b>Left</b>	Datos registrados mientras se está haciendo un ejercicio de imaginación motora en el que un sujeto imagina desplazar la mano diestra hacia la izquierda para mover un ratón.

**Tabla 8: Etiquetas de identificación de los eventos en cada database**

Por último, se ha adaptado la interface del LabVIEW para permitir a un usuario la interacción con el programa diseñado y poder realizar así el etiquetado y registro de los datos de forma manual (*Ilustración 19*).



**Ilustración 19: Interface de LabVIEW diseñada**

Por medio de los elementos implementados en la interface, a través del programa resultante, una vez éste se ejecute, se puede:

## Método Operativo

- Detener el programa clicando sobre el pulsador de *STOP*.
- Comenzar el registro de datos pulsando el interruptor *RECORD*.
- Seleccionar la etiqueta con la que se quiera identificar la señal clicando sobre la pestaña situada encima del interruptor de registro.
- Visualizar la barra que indica la calidad de contacto entre los electrodos del MUSE 2 y la cabeza del sujeto que lo emplee.
- Visualizar una gran pantalla gris con una flecha de ratón en el centro, para, tal y como se explicará más adelante, facilitar al sujeto la generación de las señales deseadas durante el proceso de registro de los datos.
- Visualizar la frecuencia a la que son recibidos los datos.

### *5.3.3. Estructura y funcionamiento del programa*

Dentro de este apartado se va a hablar de cómo está configurado el programa de LabVIEW pensado para la adquisición de datos. Tanto el esquema completo como las funciones diseñadas personalmente pueden ser visualizados dentro de los anexos, en el punto 1.1 concretamente.

Cabe destacar que la estructura del programa está dividida en dos:

- El diseño del servidor necesario para adquirir las señales por UDP y para decodificarlas del protocolo OSC.
- El diseño de la parte del programa encargada de registrar los datos dentro de *databases*.

#### *5.3.3.1. Servidor para la adquisición de datos*

El desarrollo de esta parte se ubica dentro de un único bucle "while", el cual se encuentra en constante ejecución hasta que el usuario decida pulsar el botón de STOP. Fuera del mismo, se ha introducido el conjunto de direcciones del protocolo OSC para la selección del método a utilizar. A su vez, se ha abierto el protocolo UDP con el bloque "UDP Open", seleccionando el puerto a través del cual se va a recibir la información.

Dentro del bucle, se ha implementado el bloque "UDP" read a través del cual, la lectura de los datos transmitidos se hace posible. Tal y como se ha comentado, el contenido estará codificado, siendo en este caso un conjunto de mensajes comprimidos en el formato llamado "bundle". Este debe de ser descifrado posteriormente.

Durante el tiempo que transcurre mientras no llega ningún mensaje procedente de la aplicación "Mind Monitor", se está produciendo un error (error 56) dentro de "UDP

Read", el cual indicará al programa no hacer nada. En el caso de que el error producido difiera del esperado, el programa se detendrá.

Cuando no hay ningún tipo de error, el programa considera que ha llegado un algún mensaje codificado en OSC y comprimido en "*bundles*". Para su análisis se ha diseñado una función llamada "*Bundle to Message Dispatched*", la cual, teniendo como entradas el mensaje y las direcciones OSC, emite cada una de las direcciones con sus respectivos mensajes.

Finalmente, para diferenciar cada uno de los métodos en función de la dirección, se ha utilizado una *estructura de caso*. Con cada una de las opciones, se ha diseñado una función encargada de decodificar sus respectivos contenidos, pudiéndose ver en la interface del programa a tiempo real una vez se ejecute.

### 5.3.3.2. Programa para el registro de datos

Para poder lograr el registro de los datos que conforman las señales EEG captadas, se ha tenido que idear un sistema de comunicación por colas con el cual poder transmitir información a tiempo real entre dos bucles "*while*" de forma síncrona. A su vez, para la gestión de las mismas se ha implementado un bucle adicional. En éste también se encuentra el botón de paro del programa (*STOP*), cuya activación permite la detención de los tres bucles sincronizadamente.

Dentro del método a través del cual se extrae las señales EEG, se ha implementado una *estructura de caso* que únicamente quedaría activa cuando se pulsa un botón, designado como "*Record*". En ese momento, las cuatro colas introducidas previamente comenzarían a guardar cada uno de los datos, estableciendo una comunicación con el bucle "*while*" encargado del registro.

Las variables recién transmitidas son cambiadas de tipo de dato a *string* gracias al bucle "*Format Into String*", así como información del estado de la comunicación entre los electrodos y la cabeza del sujeto. Por otro lado, con una *estructura de caso*, se ha definido una nueva variable con la etiqueta del evento que se quiera registrar en un momento determinado, que variará a manos del usuario mientras se comunica por la interace.

Finalmente, cuando el programa es detenido al pulsar *STOP*, el conjunto de datos registrados dentro de este último bucle "*while*" son agrupados en un array (matriz) de dos dimensiones. El conjunto resultante es procesado gracias al bloque "*Write Delimited Spreadsheet*", con el cual se permite al usuario guardar un fichero con el contenido entrante. Cabe destacar que si se da la orden de introducir una "coma" en el espacio que

separa cada variable a través de la entrada "delimiter(/t)", el archivo puede ser guardado en formato CSV.

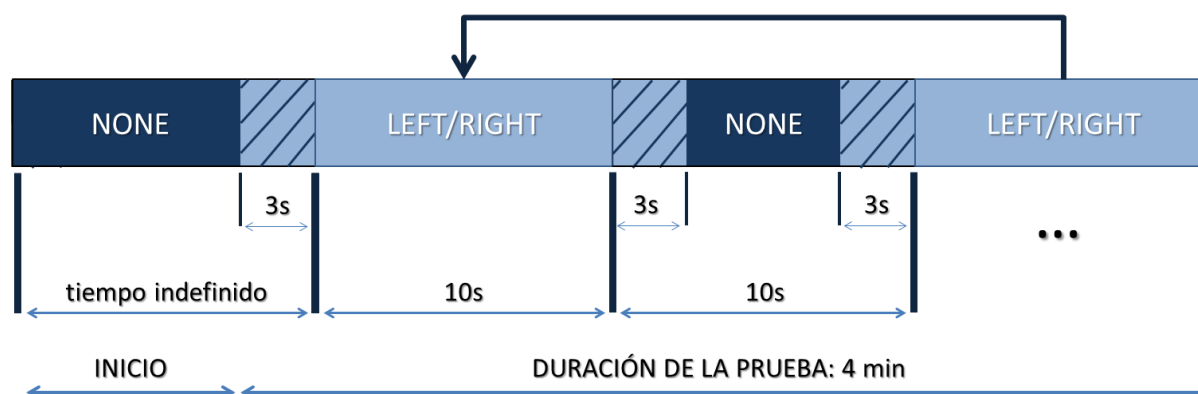
### 5.3.4. Ensayos para el registro

Para este proyecto, la realización de pruebas, con el objetivo de guardar las señales EEG dentro de un *database*, se han realizado con un único sujeto, de 21 años de edad, que no es inválido, no practica meditación de forma regular y es diestro.

En total, se ha realizado una suma de 25 grabaciones o ensayos, de las cuales, 21 han sido posteriormente utilizadas para entrenar el algoritmo de clasificación y, las 4 restantes, para el testeo del modelo resultante.

En la realización de cada uno de los ensayos, ha sido necesaria la participación de dos personas. Por un lado, el sujeto que se va a colocar el dispositivo EEG y, por el otro, una persona encargada de supervisar la prueba, así como controlar la interface diseñada en LabVIEW.

En la *Ilustración 20*, se ha intentado resumir gráficamente el funcionamiento de cada uno de los ensayos realizados. Tal y como se puede comprobar, la prueba comienza identificando los datos con la etiqueta "None". De este modo, se esperará un tiempo no definido hasta que el sujeto se concentre completamente en la tarea, identificando mientras tanto las señales registradas como no válidas para evitar errores en el entrenamiento de los modelos.



**Ilustración 20: Prueba ideada para la toma de datos**

En cada grabación, la persona a la que se le ha colocado el dispositivo debe sentarse de forma cómoda, concentrándose en la pantalla gris ubicada en la interface del programa. Ésta, en su interior, consta de cuatro rectángulos que representan las cuatro direcciones que querríamos intentar predecir (en el proyecto solo se va a hacer con dos), y una flecha, que simula ser el puntero de un ratón, situada en el centro.



Una vez el supervisor se asegura de que el dispositivo ejerce un buen contacto sobre la cabeza del sujeto, así como de que éste se encuentra calmado y evitando realizar movimientos bruscos, la prueba puede comenzar. En este momento, se activa manualmente un cronómetro para controlar de forma aproximada la duración de la prueba. El tiempo de la misma debe ser de 4 minutos más o menos, aunque en algún caso se ha decidido alargar o acortar.

Al comienzo de la prueba, el supervisor manda una orden de imaginación motora al sujeto, ya sea pensar en ejecutar un desplazamiento de mano hacia la derecha o hacia la izquierda. Sin embargo, en el instante que se dicta el evento, no se cambia la etiqueta en la pestaña de la interface, sino que se espera unos tres segundos y luego se procede a seleccionar la opción de la orden correspondiente. De este modo, uno se asegura que la señal a la cual se le ha etiquetado un evento en concreto es fiable, ya que si el cambio de etiqueta se produjera en el momento que se dicta, lo más posible es que, como mínimo, se registrara unos milisegundos de datos erróneos al inicio.

Una vez etiquetados 10 segundos de uno de los pensamientos y sin avisar al sujeto del cambio, se selecciona la pestaña de "None" para dejar de registrar el evento y, tras esperar 3 segundos, se notifica, dejando 4 segundos para descansar.

Finalmente, se dicta una nueva orden y, de nuevo, al pasar 3 segundos, esta orden comienza a etiquetarse, repitiendo en bucle los pasos realizados hasta que se considere que la prueba a finalizado.

En cada grabación, mientras tiene lugar uno de los dos eventos, es muy importante no retirar la vista del puntero que se encuentra en el centro de la pantalla gris, ni mover la mano en la realidad. De lo contrario se correría el riesgo de generar artefactos EOG o EMG. Se debe resaltar que evitar realizar un parpadeo es muy complicado, y puede ser contraproducente al impedir que el sujeto se pueda centrar en la actividad. Por ello, estos artefactos se deberán eliminar más adelante en la etapa de procesado offline.

### 5.3.5. Archivos resultantes

Tras realizar la prueba a un sujeto determinado y pulsar el botón de Stop, el programa permite el guardado de los datos en un archivo en formato CSV que va a funcionar a modo de *dataset*. Los registros guardados se encontrarán distribuidos entre las columnas resumidas en la *Tabla 9*.

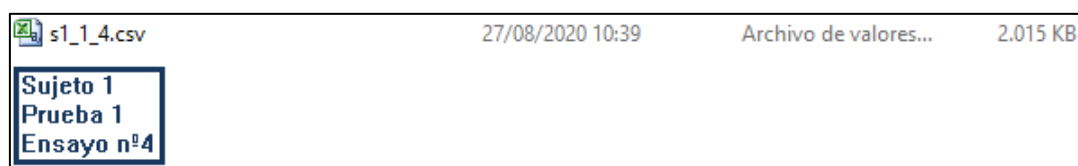
Por otro lado, el nombre de cada uno de los *datasets* se determinará en función de las siguientes normas (*Ilustración 21*):

## Método Operativo

- En el inicio se debe identificar a quien se le está haciendo la prueba. Para ello, se comienza poniendo una letra "s" seguido del número identificativo del sujeto. En este caso, al solo haber una persona, en todas las situaciones se identificará por "s1".
- Posteriormente, y separado por una "barra baja" ( \_ ), se señala numéricamente el tipo de prueba a realizar. Como en este proyecto solo hay un tipo de prueba de dos eventos (imaginar desplazamiento de mano hacia la derecha o izquierda), la única cifra que se va a escribir es "1".
- Finalmente se identifica el número de ensayo, es decir, la cantidad de veces que se ha realizado cada tipo de prueba.

Nombre de columna	Descripción
EEG_TP10	Datos de la señal EEG captados a través del canal de registro TP10
EEG_TP9	Datos de la señal EEG captados a través del canal de registro TP9
EEG_AF8	Datos de la señal EEG captados a través del canal de registro AF8
EEG_AF7	Datos de la señal EEG captados a través del canal de registro AF7
Conectivity	Calidad del contacto de los canales de registro con la cabeza del sujeto. 1 = Todos los canales están conectados. 2 = Uno, dos o tres canales están mal conectados. 4 = Todos los canales están mal conectados.
Label	Etiqueta que identifica la señal con el evento que está teniendo lugar.

**Tabla 9: Descripción del tipo de dato que posee cada columna en los databases**



**Ilustración 21: Identificación de un dataset**

Como ya se ha comentado, en total se ha grabado un máximo de 18 ensayos, a partir de los cuales se construirá la base de datos necesaria para entrenar los modelos predictivos, así como la que los testeará.

## 5.4. PYTHON

### 5.4.1. Construcción de los datasets

Para poder agrupar los 25 *databases* en un par de archivos únicamente, se ha realizado programación en *Python* [57] a través de un *Notebook* de la aplicación *Jupyter*, a la cual se puede acceder tras la descarga del navegador *Anaconda* [58].

A través de dicho programa, gracias a la librería de *Python* "*Pandas*" [59], se ha cargado cada uno de los archivos en formato CSV y, mediante la implementación de 3 nuevas columnas, se ha identificado cada uno de ellos manualmente de acuerdo con su nombre. Estas columnas mostrarán el sujeto al que le corresponde el archivo, la prueba y su número de ensayo.

De forma posterior, se ha eliminado los datos correspondientes a valores diferentes de 1 en la columna "*Conectivity*", al implicar en esos casos que alguno de los sensores, en un momento determinado, no estaba haciendo buen contacto con el sujeto. También, se ha eliminado el conjunto de datos con valores "*NaN*", es decir, las filas en las que, a causa de algún error, los datos no pudieron registrarse y pasaron a contar como nulos. Usando la función "*isnull().sum()*", se ha comprobado que no queda ningún "*NaN*" sin retirar.

Por último, se ha añadido al *database* un par de columnas adicionales para asociar cada una de las filas con un momento temporal, así como indicar la frecuencia de muestreo a la que se han registrado los datos (256 Hz).

Una vez modificados los archivos, estos han sido agrupados, dejando únicamente dos *databases*, el de entrenamiento ("*db\_1\_train.csv*") y el de testeo ("*db\_1\_test.csv*"), los cuales serán descargados y guardados en el ordenador.

### 5.4.2. Procesado de los datos

De forma previa a la construcción de un modelo, es necesario procesar los datos de las señales EEG en bruto de la forma más óptima. Para ello, la programación se ha realizado nuevamente en un *Notebook* de lenguaje *Python*, cambiando, sin embargo, de entorno al *Google Colab* [60]. De esa forma, se logra la conexión a un entorno de ejecución alojado provisto de *GPU*, lo cual permite optimizar la ejecución de algoritmos complejos, así como dejar almacenado el código en *la nube*.

### 5.4.2.1. Preparación de los datos de entrenamiento

En primer lugar, una vez instalada la librería de Pandas y habilitado el *Drive* para obtener el acceso a sus archivos, se ha procedido a la lectura de la base de datos de entrenamiento. Ésta debe de modificarse, quedándose únicamente con las columnas que contiene las señales en bruto de los canales, así como sus respectivas etiquetas, las cuales se han pasado a identificar con un número entero (*Tabla 10*).

<b>None</b>	<b>0</b>
<b>Left</b>	<b>1</b>
<b>Right</b>	<b>2</b>

**Tabla 10: Número de identificación de cada etiqueta**

De forma posterior, se va a implementar la librería *MNE* [61], la cual posee funciones predefinidas especializadas en el procesado de señales EEG. Para su correcto funcionamiento, es necesario convertir la base de datos en un objeto denominado en la documentación de la librería como *"RAW"*. Por ello se ha diseñado una función encargada de adaptar el *database* a los requerimientos necesarios para crear el objeto, la cual se ha llamado *"create\_mne\_raw\_object(data)"*.

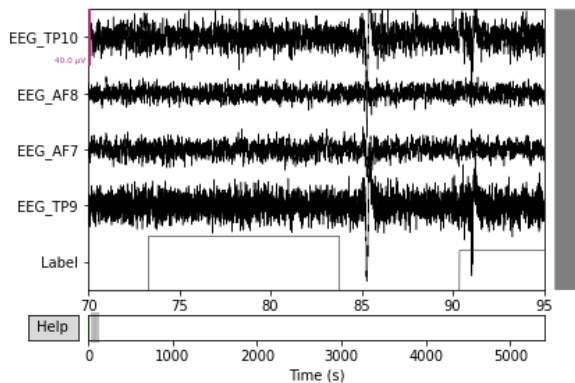
### 5.4.2.2. 1<sup>er</sup> filtro FIR paso banda

Una vez conseguido el objeto necesario para emplear la librería *MNE* (se ha llamado *"raw\_TrainDb\_s1"*), los señales pueden imprimirse con gran facilidad gracias al comando *"raw\_TrainDb\_s1.plot"*, viendo así la forma de onda que presentan en cada uno de los canales (*Ilustración 22*).

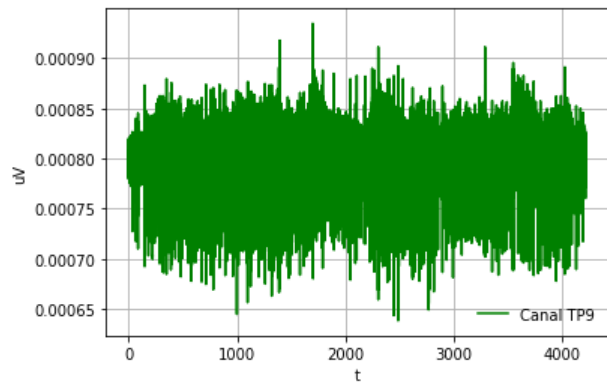
Sin embargo, de este modo, no se puede observar, en el eje *y*, el valor numérico de la tensión que las señales tienen. Por ello, se ha utilizado la librería *"matplotlib"* [62], imprimiendo la señal registrada a través del canal TP10 (*Ilustración 23*), pudiendo visualizar sus valores en la unidad de los microvoltios.

Tal y como se puede ver, el valor de la señal es demasiado alto, notándose la presencia de una tensión de offset de entre 750 y 800  $\mu\text{V}$  causada, seguramente, por el ruido infiltrado mientras se estaba realizando los ensayos.

Para detectar la presencia de ruido en las señales EEG, comúnmente se busca manifestaciones del fenómeno en el espectro de frecuencia, ya que supondría que en ciertos puntos la amplitud presentaría variaciones fácilmente detectables. Por ello se ha impreso el espectro de densidad de potencia (*PSD*) gracias al comando de la librería *MNE* *"raw\_TrainDb\_s1.plot\_psd()"*.

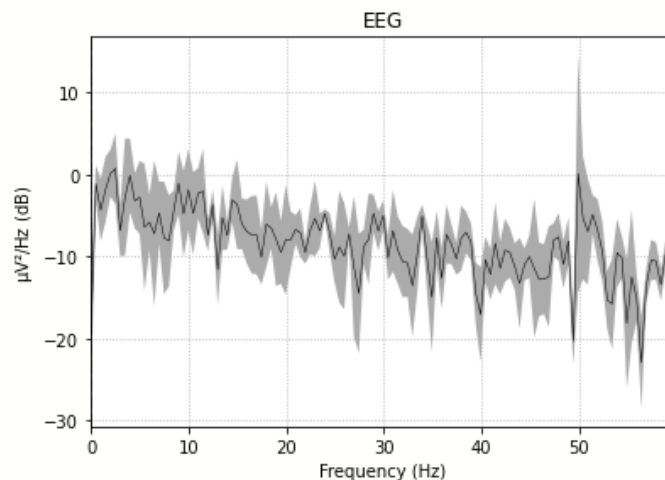


**Ilustración 22: Señal EEG en bruto**



**Ilustración 23: Señal EEG del canal TP10**

En la *Ilustración 24* se puede visualizar el promedio de la potencia de los cuatro canales, partiendo de una ventana de 2 segundos (tiempo inicial = 70s). En ella, es fácilmente reconocible un “pico” de potencia alrededor de los 50 Hz, atribuido al ruido causado por la corriente alterna instalada en los hogares (en Europa).



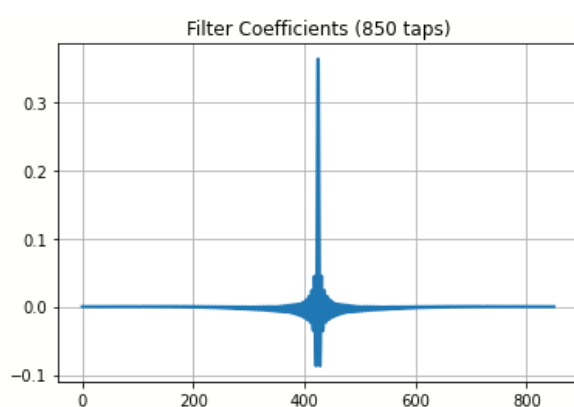
**Ilustración 24: PSD de la señal en bruto**

Lo más probable es que la tensión de offset sea causada por dicho evento. Para solucionar el problema, se ha decidido diseñar un filtro FIR de paso banda capaz de filtrar el ruido eléctrico, así como otros tipos que suelen ocurrir a frecuencias bajas como, por ejemplo, el causado por el contacto en seco entre un electrodo y la cabeza del sujeto (*Tabla 11, Ilustraciones 25 y 26*). La librería utilizada para ello ha sido SciPy [63].

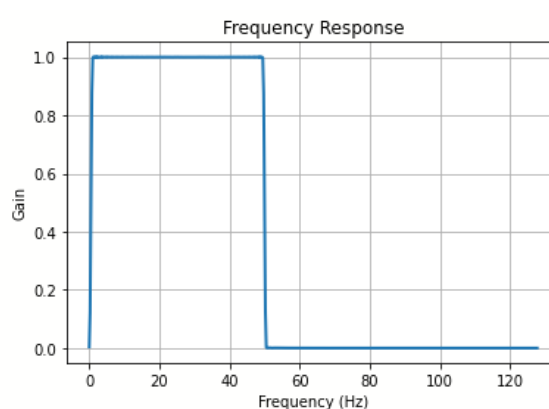
En la *Ilustración 27*, se puede visualizar el resultado final tras aplicar el filtro, siendo la señal resultante la mostrada de color azul, y la original en verde. La parte de color rojo, es la parte de la señal que se debe omitir, ya que, cuando se realiza un filtrado, se genera un retardo de fase que dependerá del número de “Taps” (*Retardo = N-1*).

Parámetro del filtro	Valor asignado
Frecuencia de Nyquist ( $f_{nyq}$ )	Frecuencia de muestreo/2 = 256/2 = 128
Frecuencias de corte	Frecuencia 1 = 0,5 Hz, Frecuencia 2 = 50 Hz
Número de "Taps" (N)	900
Ventana	"Hamming"

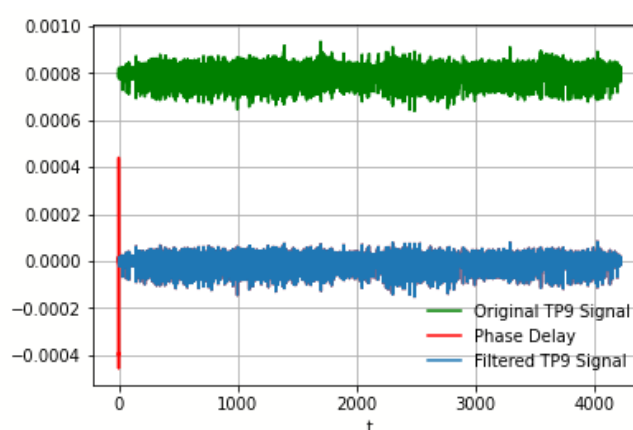
**Tabla 11: Parámetros para el diseño el filtro 1**



**Ilustración 25: Coeficientes del filtro 1**

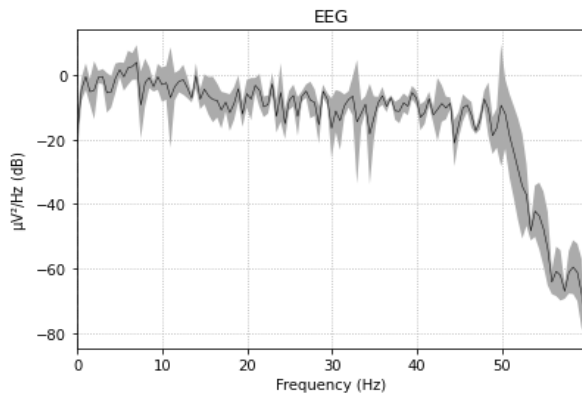


**Ilustración 26: Respuesta del filtro 1**

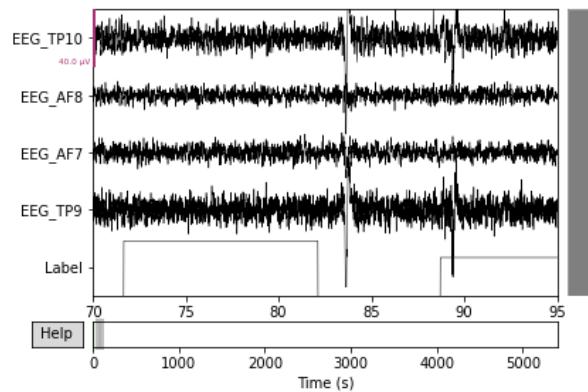


**Ilustración 27: Resultado en TP9 tras aplicar el filtro 1**

Una vez eliminado, visualizando nuevamente el espectro, se podría comprobar que la potencia existente dentro de los rangos filtrados se ha reducido en gran medida (*Ilustración 28*). También se podría imprimir los datos de salida en los cuatro canales a través de la función "raw\_TrainDb\_s1.plot", estando éstos liberados del ruido (*Ilustración 29*).



**Ilustración 28: PSD tras aplicar el filtro 1**



**Ilustración 29: Señales EEG tras aplicar el filtro 1**

### 5.4.2.3. Filtrado de artefactos

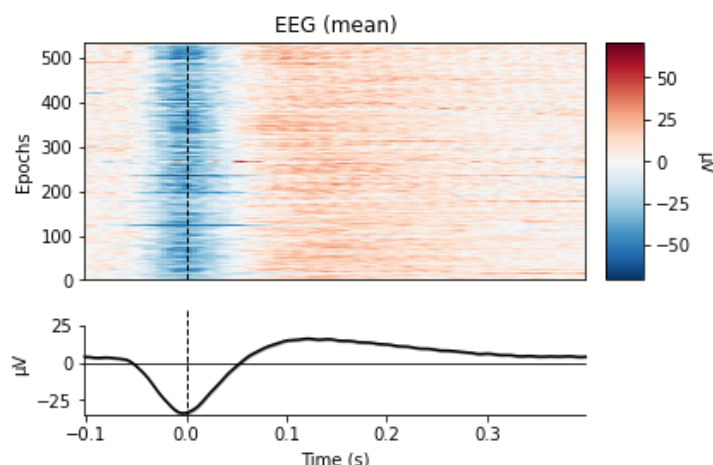
Tal y como se ha podido ver en la *Ilustración 24*, en el intervalo de los 80 y 90 segundos, indicados en el eje de X, se puede ver una variación brusca de la señal. Ésta se trata de una interferencia causada por un parpadeo del sujeto al que se le estaba realizando el ensayo, es decir, un artefacto categorizado dentro de los EOG.

Gracias a la librería de Python *MNE* [61], se ha hecho uso de la función prediseñada "*mne.preprocessing.create\_eog\_epochs()*" con objeto de detectar los tramos, llamados "*epochs*" ("*épocas*" en castellano), donde ha podido tener lugar un artefacto de este tipo. Para ello, la función detecta cambios en cuanto a la correlación de la señal, ya que cuando el sujeto parpadea, la tensión desciende de forma elevada en poco tiempo y, posteriormente experimenta un incremento.

Los tramos se han guardado dentro de la variable "*eog\_epochs*" para poder ser impresos mediante la ejecución del comando "*eog\_epochs.plot\_image(combine='mean')*" (*Ilustración 30*).

Tal como ya se ha dicho previamente, a la hora de registrar los datos es importante evitar la generación de este tipo de eventos. Sin embargo, por ejemplo, parpadear es inevitable y, por ello, estos artefactos deben ser eliminados durante el procesado.

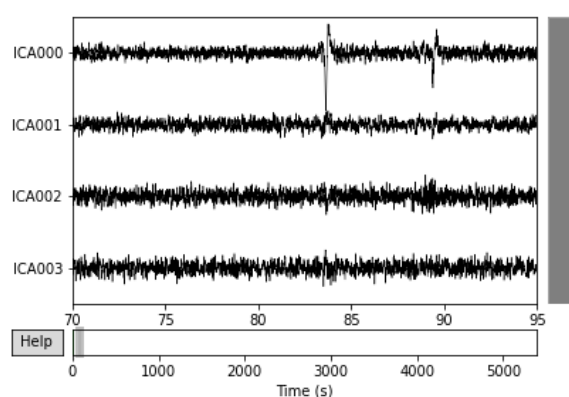
Para ello hay dos opciones, eliminar las porción de la señal vinculada a los EOG o intentar filtrarlas gracias a diversos algoritmos, previamente mencionados en el marco teórico. Como el presente trabajo se está realizando con la motivación de que en un futuro sirva para crear una BCI, en éstas el procesado es a tiempo real y, por lo tanto, no se puede optar por la primera de las vías. En consecuencia, se ha optado por la opción de filtrar los artefactos a través de la técnica de Análisis de Componentes Independientes (ICA), implementada en la librería MME.



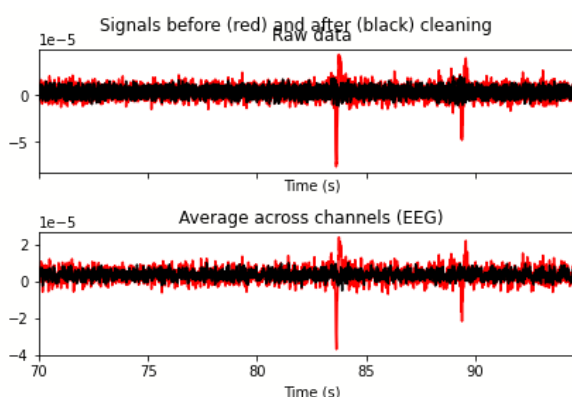
**Ilustración 30: Correlación de la señal ante eventos de EOG**

En la *Ilustración 31*, se puede visualizar los cuatro componentes independientes (ICs) extraídos por la técnica ICA (tramo de la señal:  $t_i = 70s$ ,  $t_f = 95s$ ). Para lograrlo, el programa ha utilizado el algoritmo denominado "Picard", uno de los más recientemente diseñados, rápidos y efectivos [9].

Tal y como se puede observar, los artefacto EOG causados por el parpadeo son fácilmente visibles en el componente ICs denominado "ICA000". El cambio causado tras la retirada del componente puede verse en la *Ilustración 32*, siendo la señal de color rojo la de entrada y de negro la de salida.



**Ilustración 31: ICs**

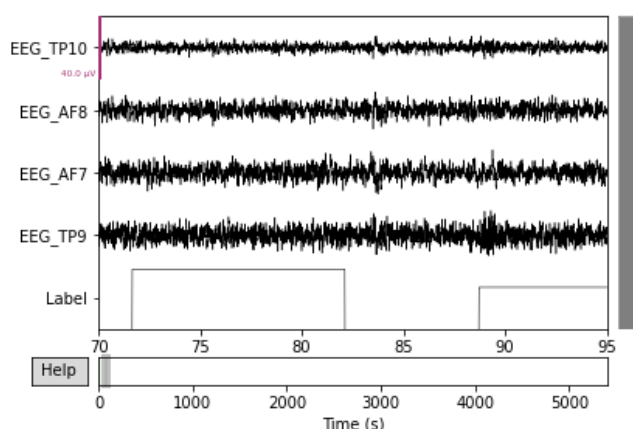


**Ilustración 32: Resultados tras emplear el ICA**

#### 5.4.2.4. 2º filtro FIR paso banda

Tras la retirada de los artefactos EOG, las señales resultantes en cada uno de los cuatro canales serían las mostradas en la *Ilustración 33*. Teniendo ahora los datos más limpios, uno puede pasar al punto del procesado destinado a extraer las características necesarias para entrenar un clasificador.





**Ilustración 33: Señales EEG tras aplicar la técnica de ICA**

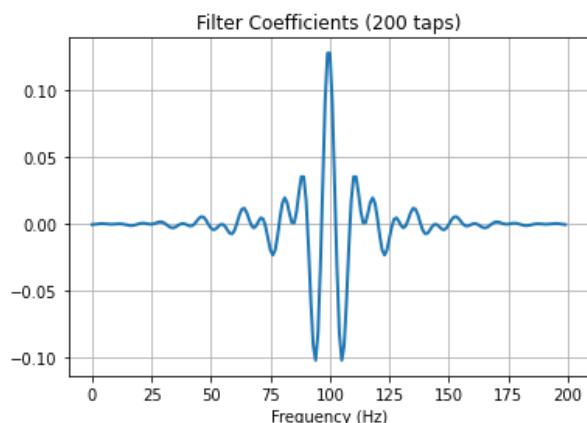
Por ello, primero se ha implementado un nuevo filtro FIR de paso banda, encargado de la eliminación de las frecuencias ubicadas en bandas no relacionadas con actividades de imaginación motora. Pese a que, generalmente son utilizadas tanto la banda Alpha como la Beta, finalmente solo se utilizará en el proyecto esta última a causa de haber dado unos mejores resultados.

Para desarrollar el filtro, el procedimiento a seguir ha sido idéntico al del anterior, en primer lugar, definiendo sus parámetros para después aplicarlo sobre la señal y eliminar el retardo de fase generado. Hay que tener en cuenta que, dentro del espectro, la banda Beta abarca desde los 13 hasta los 30 Hz.

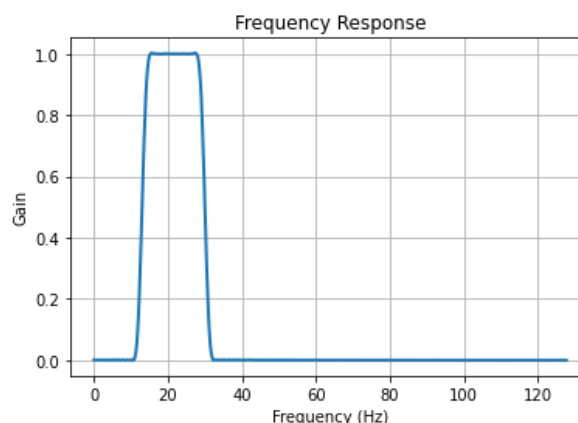
En la *Tabla 12* y en las *Ilustraciones 34 y 35*, se muestra las características del nuevo filtro diseñado y, en la *Ilustración 36*, la señal resultante del canal *TP9*.

Parámetro del filtro	Valor asignado
Frecuencia de Nyquist ( $f_{nyq}$ )	Frecuencia de muestreo/2 = 256/2 = 128
Frecuencias de corte	Frecuencia 1 = 13 Hz, Frecuencia 2 = 30 Hz
Número de "Taps" (N)	200
Ventana	"Hamming"

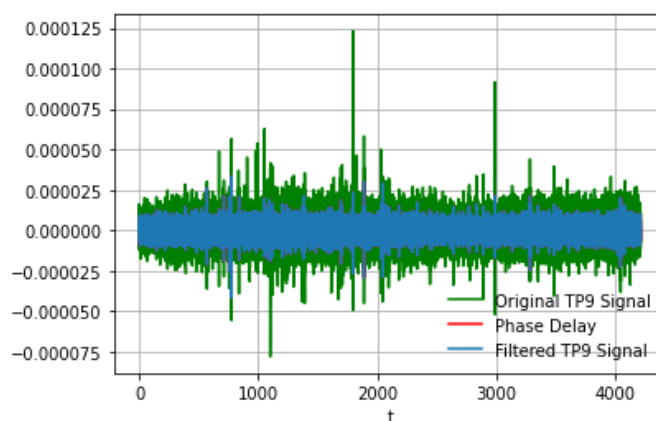
**Tabla 12: Parámetros para el diseño el filtro 2**



**Ilustración 34: Coeficientes del filtro 2**



**Ilustración 35: Respuesta del filtro 2**



**Ilustración 36: Resultado en TP9 tras aplicar el filtro 2**

#### 5.4.2.5. Filtro CSP

Dentro de este proyecto, para lograr la extracción de características, al trabajar solo con dos eventos (más la etiqueta "None"), se ha optado por el uso de filtros espaciales CSP. Tal y como se ha dicho en el apartado teórico, éstos tienen la labor de aumentar los valores relacionados con una de las etiquetas ("Left") y reducir los de la otra ("Right").

De forma previa al entrenamiento del filtro, se ha eliminado la señal de los canales TP10 y AF8, ya que se ha considerado que su contenido es mayormente ruido. Esto es debido a que, como el sujeto es diestro, el control de la mano derecha está altamente correlacionado con el lado izquierdo del cerebro, zona donde se encuentran los otros dos canales, TP9 y AF7 (la hipótesis se ha comprobado en el proyecto personalmente).

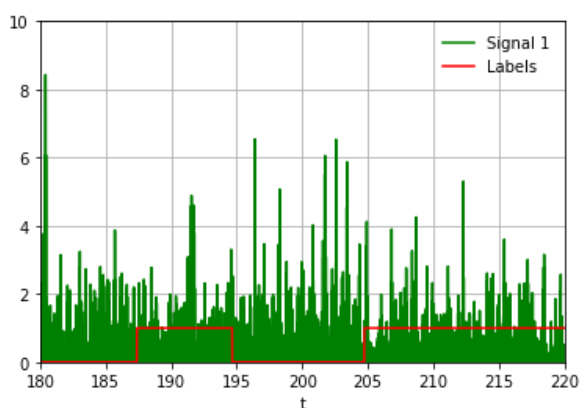
Tras reducir la dimensión de los datos, en primer lugar, se ha utilizado el comando "find\_events()" para identificar los puntos en los cuales se produce un evento de "Left" o "Right", guardándose dentro de una variable. De esta forma, gracias a la función

"Epochs()", se ha realizado una lista con las señales EEG asociadas a cada uno de los eventos en los tramos desde 0.5 hasta 2.5 segundos, empezando a contar cuando éstos comienzan a etiquetarse. A través de la biblioteca *Numpy* [65], también se ha guardado, en un vector *array* aparte, una lista con las respectivas etiquetas de cada uno de los tramos, siendo "1" el número identificativo asociado al evento "Left" y, "-1", el correspondiente a "Right".

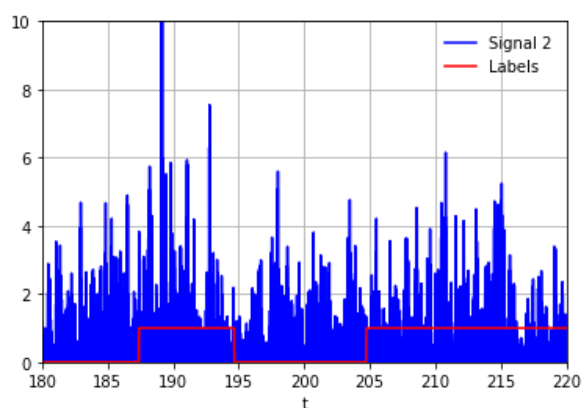
Posteriormente, se entrena el CSP, seleccionando de forma previa el número de filtros a diseñar, que, en este caso, es de dos (no se puede superar al número máximo de canales de registro).

Una vez diseñado, se ha editado el objeto "raw\_TrainDb\_s1" para retirar de los datos las partes asociadas a la etiqueta "None", quedándose solo el contenido con el cual se va a entrenar el clasificador para predecir los dos eventos (Left y Right).

Tras hacerse, se procede a aplicar los filtros CSP sobre las señales de los canales de registro no eliminados, obteniendo como resultando dos *arrays* unidimensionales. A su vez, los valores resultantes son rectificadas elevándolos al cuadrado (Ilustraciones 37 y 38), eliminando así los componentes negativos.



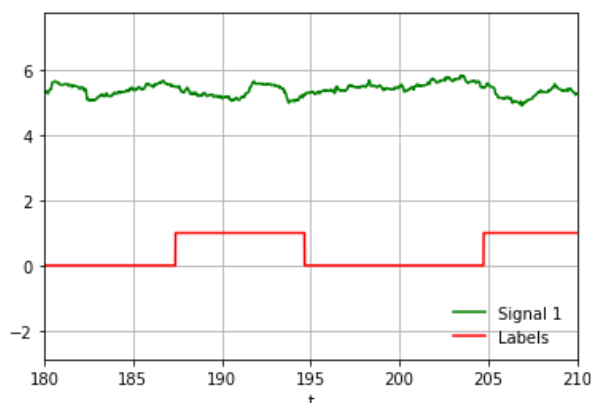
**Ilustración 37: Señal 1 al aplicar el CSP**



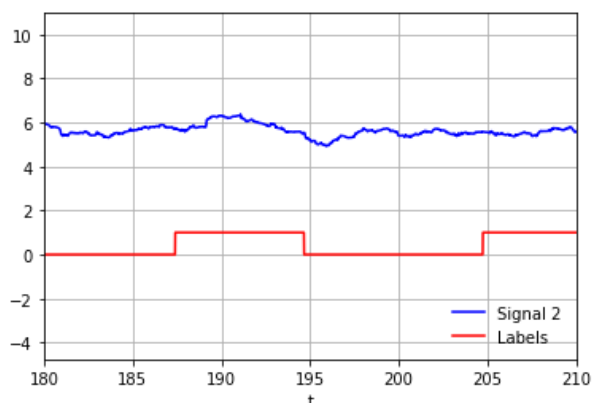
**Ilustración 38: Señal 2 al aplicar el CSP**

Para finalizar el procesado, se realiza un proceso de alisado de la señal gracias a una serie de convoluciones que funcionan a partir de ventanas rectangulares, cuyo tamaño se ha decidido que sea de unas 800 muestras. Posteriormente, se ha aplicado un logaritmo neperiano sobre los resultados para poder reducir la magnitud de los mismos (Ilustraciones 39 y 40).

De este modo, al final ha quedado una variable denominada "train\_result", que contiene un vector con el conjunto de los datos orientados al entrenamiento, así como "labels\_train", *array* con las etiquetas correspondientes.



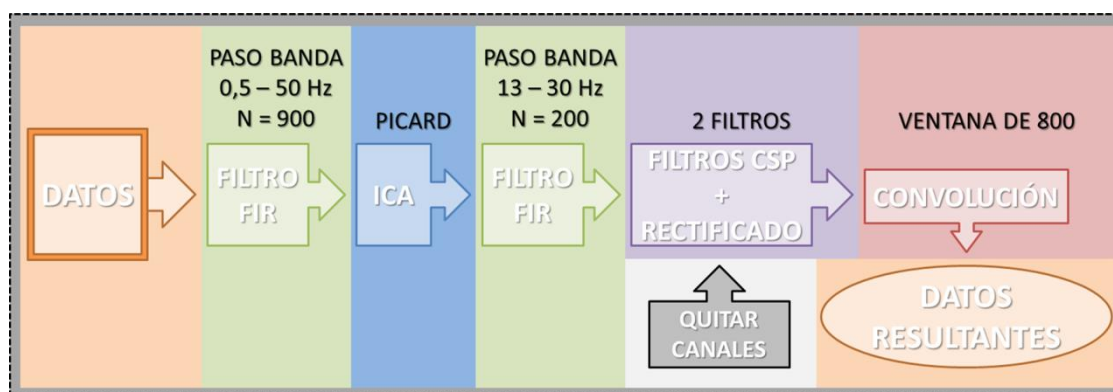
**Ilustración 39: Señal Resultante 1 (Train)**



**Ilustración 40: Señal Resultante 2 (Train)**

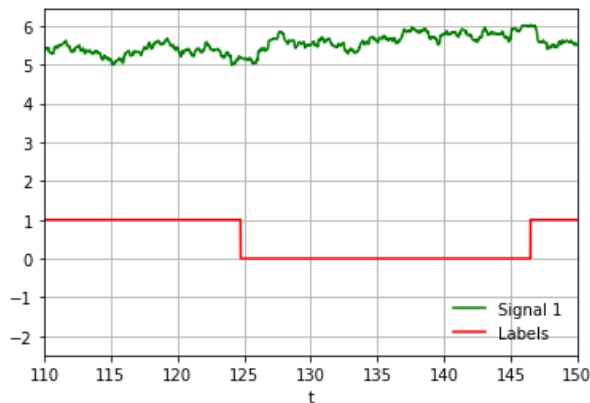
### 5.4.3. Procesado del *database* de testeo

Usando el mismo conjunto de técnicas que se ha empleado durante el procesamiento de los datos de entrenamiento, se debe repetir esta etapa con los contenidos del *database* de testeo. Para ello, no se debe reajustar ningún filtro o algoritmo, ya que el procesamiento de los datos se tiene que hacer por medio de los elementos ya diseñados, pudiéndose resumir tal y como se muestra en la *Ilustración 41*.

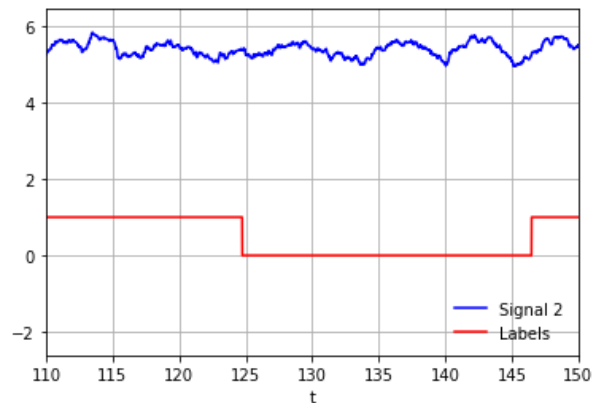


**Ilustración 41: Procesado de la Señal**

Tras aplicar el conjunto de técnicas mostradas, al igual que en el caso de los datos de entrenamiento, se ha logrado obtener como resultado una variable *numpy* que contiene una matriz de dos filas con los datos resultantes tras haber extraído las características (*Ilustraciones 42 y 43*). A su vez, también se ha guardado el array "*labels\_test*" con las etiquetas correspondientes a cada uno de los datos.



**Ilustración 42: Señal Resultante 1 (Test)**

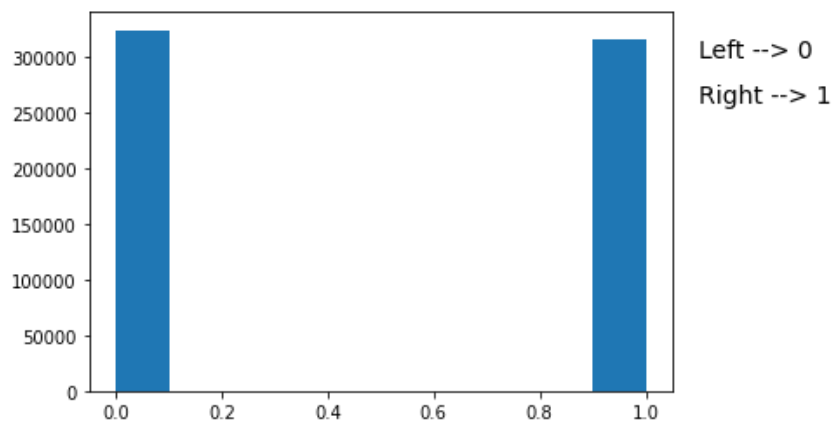


**Ilustración 43: Señal Resultante 2 (Test)**

#### 5.4.4. Clasificación de características

Por medio del conjunto de datos resultante tras la etapa de extracción de las características, se ha procedido con el diseño del modelo predictivo capaz de clasificarlas.

Para ello, en primer lugar, se debe tener muy presente el número de etiquetas con el que se trabaja, ya que, si se entrena una característica mucho más que la otra, se corre el riesgo de desbalancear el resultado, acertando siempre en una pero fallando en la otra. Para asegurar el equilibrio, se ha impreso una gráfica con las etiquetas del conjunto de entrenamiento que puede verse en la *Ilustración 44*.



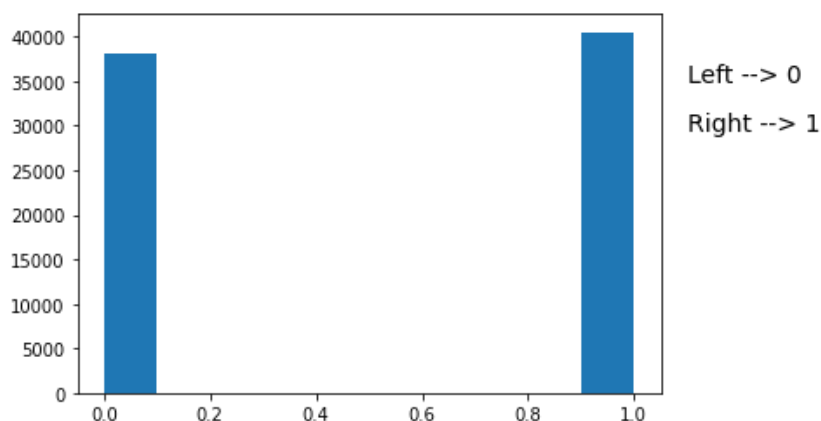
**Ilustración 44: Etiquetas totales para el entrenamiento**

Al tener una cantidad parecida, se ha podido pasar al diseño de modelo. Éste debe de entrenarse a través de los datos de la variable *Train* para, posteriormente, evaluar su capacidad de predicción por medio de los contenidos en *Test*.

El conjunto total de etiquetas asociadas a los datos de testeo pueden verse en la *Ilustración 45*, siendo éstas las que el clasificador debe predecir para que éste sea considerado válido.

## Método Operativo

Tal y como se ha dicho en el apartado teórico, los algoritmos a emplear se pueden dividir en dos grandes grupos, *lineales* y *no lineales*. En el trabajo actual finalmente se ha construido dos modelos predictivos partiendo de las técnicas de *Regresión Logística* y *XGboost*, ambas de clasificación *lineal*. Para poder implementar los algoritmos, se ha utilizado la librería “*Scikit-Learn*” (*sklearn*) [65], especializada en funciones que faciliten la utilización de técnicas de *machine learning* como las ya citadas.



**Ilustración 45: Etiquetas totales para el testeo**

Tras entrenar el modelo, por medio de los datos de entrenamiento, y testearlo al aplicarlo sobre el conjunto de testeo, se ha comprobado el porcentaje de acierto gracias al comando “*accuracy\_score()*”. Sin embargo, este resultado no es capaz de aportar unas conclusiones consistentes, ya que hay que ver la distribución de los eventos clasificados. Por ello es importante la visualización de la matriz de confusión.

Toda esta información ha sido anotada y analizada en la sección de “*Resultados*”, la cual se encuentra en el capítulo siguiente.

## 6. RESULTADOS

### 6.1. CAPACIDAD DE I.M. DEL SUJETO

En el presente Trabajo de Fin de Grado, los ensayos han sido realizados a un único sujeto, que ha tenido que realizar ejercicios de imaginación motora para poder generar una señal de respuesta en su cerebro.

Tal y como se ha comentado en el apartado 4.3, el MIQ-R es uno de los cuestionarios más completos en este campo y, por lo tanto, con éste se ha decidido valorar la habilidad de imaginación motora del sujeto.

Una vez realizado el cuestionario, los resultados obtenidos se han anotado en la Tabla 13.

RESULTADOS DEL CUESTIONARIO REVISADO DE IMAGEN DEL MOVIMIENTO [50]			
Tipo de Prueba	Número de Prueba	Resumen del movimiento	Puntuación
Imagen Cinestésica	Ítem 1	Levantamiento de la rodilla derecha manteniendo la otra en el suelo	4
Imagen Visual	Ítem 2	Inclinación hacia abajo y salto con brazos extendidos sobre la cabeza	6
Imagen Cinestésica	Ítem 3	Desplazamiento vertical del brazo de la mano no dominante	7
Imagen Visual	Ítem 4	Movimiento de cintura intentando tocar los pies con los dedos	5
Imagen Cinestésica	Ítem 5	Inclinación hacia abajo y salto con brazos extendidos sobre la cabeza	5
Imagen Visual	Ítem 6	Levantamiento de la rodilla derecha manteniendo la otra en el suelo	7
Imagen Cinestésica	Ítem 7	Movimiento de cintura intentando tocar los pies con los dedos	4
Imagen Visual	Ítem 8	Desplazamiento vertical del brazo de la mano no dominante	6

**Tabla 13: Resultados del MIQ-R**

## Resultados

De forma posterior, se ha calculado la media, estableciendo una separación entre las dos subescalas:

$$\text{Imagen Cinestésica} = \frac{4 + 7 + 5 + 4}{4} = 5$$

$$\text{Imagen Visual} = \frac{6 + 5 + 7 + 6}{4} = 6$$

Tal y como se puede ver, según los valores obtenidos y de acuerdo con la escala de las puntuaciones, así como la interpretación ofrecida por el cuestionario [52], se podrían sacar las siguientes conclusiones:

- El sujeto tiene facilidad a la hora de formar la imagen de movimiento.
- El sujeto siente ligeramente el movimiento cuando se lo imagina.

Por consiguiente, se podría considerar que los resultados no son malos, teniendo el sujeto una mayor capacidad de formar la imagen de movimiento que de sentirlo.

Para finalizar, se ha considerado conveniente matizar que cada una de las pruebas ha sido repetida en varias ocasiones hasta poder dar una valoración oportuna. Se ha podido apreciar que cuantas más veces se reiniciara la tarea, ésta era más sencilla de realizar.

## 6.2. EVALUACIÓN DE LOS CLASIFICADORES

Tal y como se ha mencionado previamente, dentro del trabajo se han diseñado varios modelos a partir de distintas técnicas que, imprimiendo su exactitud mediante el comando "*accuracy\_score()*", han dado resultados variados, anotados en la *Tabla 14*.

ALGORITMO	PRECISIÓN EN LA PREDICCIÓN DE <i>TRAIN</i>	PRECISIÓN EN LA PREDICCIÓN DE <i>TEST</i>
Regresión Lineal (LR)	61,78%	68,35%
XGboost	63,53%	62,84%

**Tabla 14: Exactitud de cada modelo al aplicarlo sobre *Train* y *Test***

Estas predicciones se han realizado tanto sobre el *database* de testeo como sobre el de entrenamiento, pudiendo así comprobar que no varían en gran medida los resultados entre ambas bases de datos (<7% con Regresión Lineal y <1% con XGboost).

A continuación se va a analizar e interpretar cada uno de los modelos diseñados.



## 6.2.1. Modelo de Regresión Logística

### 6.2.1.1. Valoración de la exactitud o "accuracy"

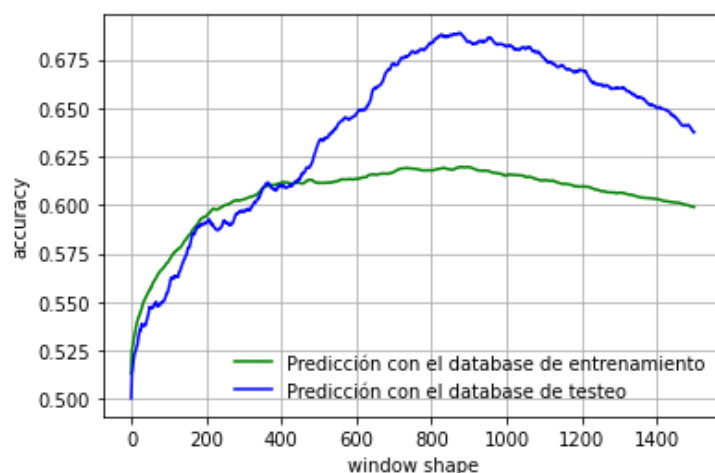
El algoritmo de Regresión Lineal ha logrado resultados con una exactitud superior al 60%, sin variar más de un 7% entre el *database* de *Test* y *Train*.

Para poder ver la evolución de la capacidad predictiva del modelo, en función del tamaño de ventana aplicada durante la etapa de alisado de la señal, se ha realizado una gráfica, mostrando la exactitud lograda sobre el eje "y" (*Ilustración 46*).

Tal y como se puede ver, mediante ventanas de tamaño pequeño, el modelo falla alrededor del 50% de las veces, es decir, al ser un clasificador binario (dos clases) lo más seguro es que esté prediciendo por azar y no porque se esté detectando un patrón. Por otro lado, se puede ver como a medida que se aumenta el tamaño de ventana, la extracción de características es más efectiva en los dos casos, llegando hasta el punto más elevado en las casi 900 muestras para, posteriormente, comenzar a decrecer.

A la hora de comparar los resultados sobre los dos *datasets* a los cuales se les ha aplicado el modelo, se puede apreciar diferencias a lo largo de su evolución.

- Al utilizar el clasificador con los datos de entrenamiento, la variación entre los resultados resulta mucho más estable al no presentar cambios bruscos cuando se hace variar el número de muestras de la ventana.
- Por el otro lado, con los datos de testeo, los resultados son más inestables, así como mostrando un resultado inferior al obtenido con los datos de *Train* al utilizar ventanas de 400 muestras o menos. En cambio, a raíz de ese punto las predicciones serán mayores.



**Ilustración 46: Evolución de la predicción con LR en función del tamaño de ventana usado en el alisado**

## Resultados

Estas diferencias son esperables, al tener el primero de los *datasets* una mayor cantidad de datos, así como ser éste el encargado de entrenar el clasificador

### 6.2.1.2. Matrices de Confusión

Para poder analizar de mejor manera la capacidad predictiva del modelo, se ha extraído la matriz de confusión de ambos resultados (*Tablas 15 y 16*).

MATRIZ DE CONFUSIÓN MODELO RL APLICADO A LOS DATOS TRAIN			
	Característica Predicha: 0	Característica Predicha: 1	Total
Característica Real: 0	205134	119365	324499
Característica Real: 1	125222	190277	315499
Total	330356	309642	639998

**Tabla 15: Matriz de confusión al aplicar el modelo RL sobre Train**

MATRIZ DE CONFUSIÓN MODELO RL APLICADO A LOS DATOS TEST			
	Característica Predicha: 0	Característica Predicha: 1	Total
Característica Real: 0	24186	13879	38065
Característica Real: 1	10988	29524	40512
Total	35174	43403	78577

**Tabla 16: Matriz de confusión al aplicar el modelo RL sobre Test**

Los coeficientes extraídos a partir de la matriz del *dataset* de entrenamiento son los siguientes:

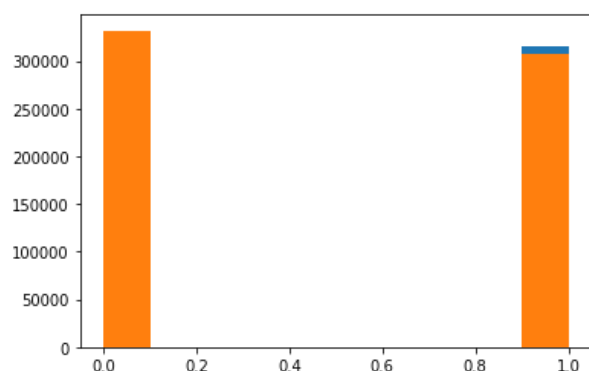
- $C_{0,0} = 205134$
- $C_{0,1} = 119365$
- $C_{1,0} = 125222$
- $C_{1,1} = 190277$

A su vez, en el caso de la que procede del dataset de testeo, los coeficientes que la forman serían:

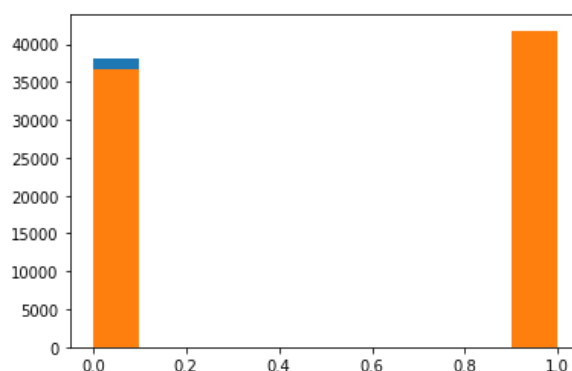
- $C_{0,0} = 24186$
- $C_{0,1} = 13879$
- $C_{1,0} = 10988$
- $C_{1,1} = 29524$

Observándolos, se puede ver que los resultados son bastante equilibrados, con más aciertos que fallos tanto en la característica "Izquierda" como en "Derecha".

El balance puede comprobarse visualmente por medio de la impresión gráfica de los valores predichos sobre las gráficas de barra que se mostraron de forma previa en las *Ilustraciones 44 y 45*. Tal y como se ha representado en las *Ilustraciones 47 y 48*, el número de predicciones de cada una de las características está correctamente balanceado.



**Ilustración 47: Predicciones del modelo RL aplicado a los datos Train**



**Ilustración 48: Predicciones del modelo RL aplicado a los datos Test**

Aunque ya se conozca el resultado, por comprobar la validez de los coeficientes, se ha realizado el cálculo manual del parámetro  $p$ , o exactitud del modelo, a partir de los coeficientes extraídos:

$$p = \frac{\sum C_{i=j}}{N} = \frac{205134 + 190277}{205134 + 119365 + 125222 + 190277} = 0.6178$$

## Resultados

$$p = \frac{\sum C_{i=j}}{N} = \frac{24186 + 29524}{24186 + 13879 + 10988 + 29524} = 0.6835$$

Éste ha dado unos porcentajes de acierto esperados, idénticos a los anotados en la *Tabla 14*.

Tras comprobar que los datos son correctos se puede proceder al cálculo y análisis de la precisión, la exhaustividad, el valor F1 y el coeficiente Kappa de Cohen aplicados sobre los valores de la matriz de confusión del *dataset* de testeo. De este modo se va a poder evaluar los resultados numéricamente.

### 6.2.1.3. Precisión

Haciendo uso de las ecuaciones propuestas en el apartado 4.2.5, se ha procedido al cálculo de la precisión tanto de la categoría "Left" como de "Right".

$$precisión_0 = \frac{TN}{TN + FN} = \frac{24186}{24186 + 10988} = 0.6876$$

$$precisión_1 = \frac{TP}{TP + FP} = \frac{29524}{29524 + 13879} = 0.6802$$

Tal y como se puede comprobar, los resultados son muy parecidos entre sí, así como similares al valor de la exactitud. De este modo, se puede demostrar de forma numérica que el modelo es capaz de predecir con una precisión similar ambas categorías.

### 6.2.1.4. Exhaustividad

Por otro lado, aplicando la fórmula para el cálculo de la exhaustividad, se ha verificado porcentualmente el número de veces que se ha acertado cada una de las categorías en función de las que se debería haber logrado.

$$exhaustividad_0 = \frac{TN}{TN + FP} = \frac{24186}{24186 + 13879} = 0.6354$$

$$exhaustividad_1 = \frac{TP}{TP + FN} = \frac{29524}{29524 + 10988} = 0.7288$$

Los resultados dejan claro que el modelo ha sido capaz de predecir correctamente un mayor número de veces el evento "derecha" que "izquierda", con una diferencia porcentual del 10% aproximadamente.

### 6.2.1.5. Valor F1

Para poder valorar el modelo a la hora de realizar predicciones de cada una de las categorías con una medida que incorpore tanto la valoración de la precisión como la de la exhaustividad, se ha recurrido al cálculo del *valor F1*.

$$F1_0 = 2 \cdot \frac{\text{precisión}_0 \cdot \text{exhaustividad}_0}{\text{precisión}_0 + \text{exhaustividad}_0} = 2 \cdot \frac{0.6876 \cdot 0.6354}{0.6876 + 0.6354} = 0.6604$$

$$F1_1 = 2 \cdot \frac{\text{precisión}_1 \cdot \text{exhaustividad}_1}{\text{precisión}_1 + \text{exhaustividad}_1} = 2 \cdot \frac{0.6802 \cdot 0.7288}{0.6802 + 0.7288} = 0.703$$

De este modo, se puede concluir en que el modelo funciona ligeramente mejor a la hora de acertar los positivos frente a los negativos, con una puntuación superior al 0.65 en ambos casos.

### 6.2.1.6. Coeficiente Kappa de Cohen

Pese a que el porcentaje de acierto se encuentra equilibrado y es superior al 65%, hay que tener en cuenta que, al ser un clasificador binario, con una exactitud hipotética del 0.5 cada uno de los aciertos podría ser perfectamente debido al azar. Por ello, tal y como se ha explicado en el punto 4.2.5.2.6, se va a calcular el Coeficiente Kappa de Cohen, el cual va a cuantificar la probabilidad de aciertos que pueden haberse realizado aleatoriamente.

En primer lugar se ha calculado el parámetro  $p_0$ , o proporción esperada en caso de que el modelo funcionara únicamente por azar:

$$p_0 = \frac{\sum C_{i,:} \cdot C_{:,j}}{N^2}$$

$$p_0 = \frac{(24186 + 10988) \cdot (24186 + 13879) + (13879 + 29524) \cdot (10988 + 29524)}{78577^2} = 0.5016$$

El resultado indica que, en caso de que se obtuviera una precisión del 0.5016, el clasificador sería completamente inútil al no ser capaz de detectar patrones en los datos.

Para poder evaluar el buen funcionamiento del mismo, finalmente se ha realizado el cálculo de  $k$ , para el cual, ya se conoce cuánto vale  $p$ :

$$k = \frac{p - p_0}{1 - p_0} = \frac{0.6835 - 0.501}{1 - 0.501} = 0.365$$

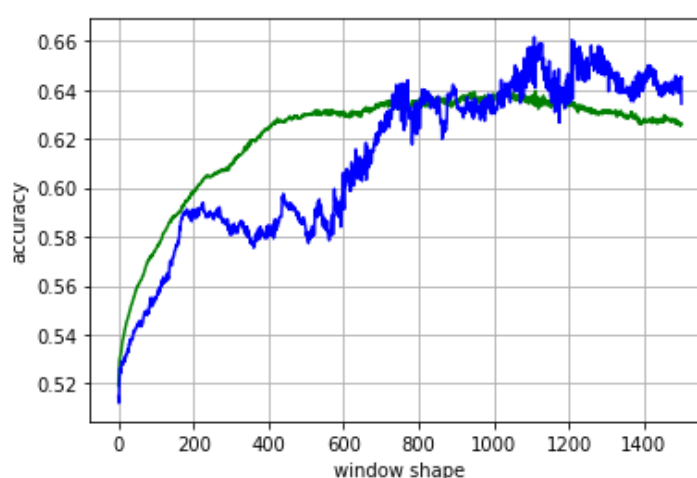
El resultado final es del 0.365, una estimación del grado de acuerdo baja pero que, sin embargo, está cerca de pasar a ser moderada. El valor indica que en un 36,5% de las ocasiones en las que el modelo acierta, éste lo hace detectando un patrón, y no al azar.

## 6.2.2. Modelo de XGboost

### 6.2.2.1. Interpretación del porcentaje de precisión

El porcentaje de exactitud al clasificar los datos con XGboost, tanto al aplicar el modelo sobre *Test* como *Train*, es superior al 60%, siendo la diferencia entre ambos más baja que en el caso del modelo generado por Regresión Logística al solo cambiar un 1%.

En la *Ilustración 49* se va a mostrar la evolución de los resultados del modelo en función del tamaño de ventana durante el alisado.



**Ilustración 49: Evolución de la predicción con XGboost en función del tamaño de ventana usado en el alisado**

Al igual que en el caso anterior, la precisión sobre el *dataset* de Train varía de forma estable sin cambios bruscos, creciendo hasta cierto punto a partir del cual el modelo pierde capacidad de acierto.

Por el otro lado, la clasificación de los datos de testeo presenta una mayor inestabilidad que cuando se aplica el modelo de Regresión Logística. En este caso, con muestras menores a 200, se logra un resultado muy inferior obtenido con el *dataset* de Train, igualándolo con 700 y superándolo ligeramente a partir de las 1000.

Pese a lograr un resultado inferior al obtenido con RL, la interpretación del modelo construido por XGboost, hasta el momento, no podría ser considerada negativa. El valor de la exactitud ronda una puntuación del 0.64 y los resultados no varían demasiado al compararse entre el *dataset* de Test y Train, traduciéndose en una mayor fiabilidad.

Sin embargo, el modelo se debe mirar con una mayor atención, ya que, como se ha visto previamente, el valor de la exactitud es un parámetro muy vago a la hora de sacar conclusiones, pudiendo estar los resultados desbalanceados y siendo fruto del azar.

### 6.2.2.2. Matrices de Confusión

Tal y como se ha hecho en el caso anterior, se han extraído las matrices de confusión con objeto de obtener unos resultados más detallados.

Una vez que éstas son comprobadas (*Tablas 17 y 18*), podemos ver una diferencia notable con el modelo RL al ver que los aciertos de la característica "Izquierda" son muy elevados, pero que, sin embargo, con "Derecha" se asimilan a los *Falsos Negativos*.

MATRIZ DE CONFUSIÓN			
MODELO XGBOOST APLICADO A LOS DATOS TRAIN			
	Característica Predicha: 0	Característica Predicha: 1	Total
Característica Real: 0	233559	90940	324499
Característica Real: 1	142573	172926	315499
Total	376132	263866	639998

**Tabla 17: Matriz de confusión al aplicar el modelo XGboost sobre Train**

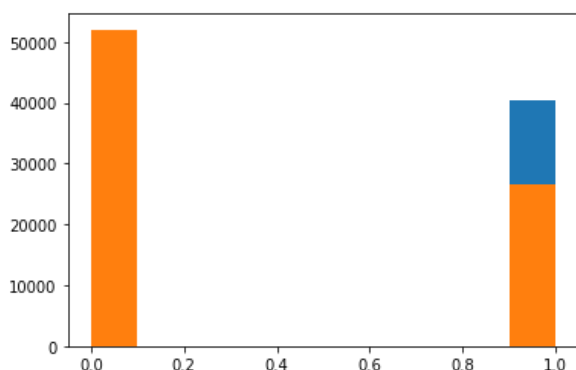
MATRIZ DE CONFUSIÓN			
MODELO XGBOOST APLICADO A LOS DATOS TEST			
	Característica Predicha: 0	Característica Predicha: 1	Total
Característica Real: 0	27137	10928	38065
Característica Real: 1	18344	22168	40512
Total	45481	33096	78577

**Tabla 18: Matriz de confusión al aplicar el modelo XGboost sobre Test**

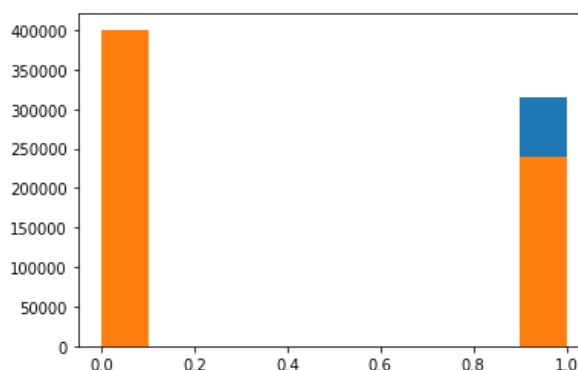
## Resultados

Este hecho se hace mucho más evidente cuando queda representado de forma gráfica (*Ilustración 50 y 51*), viéndose un importante desbalanceo en cuanto a la distribución de cada uno de los eventos predichos en comparación su reparto real.

La observación no es positiva, ya que da a entender que el modelo está demasiado entrenado en una de las clases, prediciéndola muy bien, y, en cambio, la otra apenas es capaz de detectarla.



**Ilustración 50: Predicciones del modelo XGboost aplicado a los datos Train**



**Ilustración 51: Predicciones del modelo XGboost aplicado a los datos Test**

En todo caso, los coeficientes extraídos de la matriz de confusión al aplicar el modelo sobre los datos de entrenamiento son los siguientes:

- $C_{0,0} = 233559$
- $C_{0,1} = 90940$
- $C_{1,0} = 142573$
- $C_{1,1} = 172926$

Por otro lado, los correspondientes a la matriz de confusión que resulta con los datos de testeo son los siguientes:

- $C_{0,0} = 27137$
- $C_{0,1} = 10928$
- $C_{1,0} = 18344$
- $C_{1,1} = 22168$

A partir de estos coeficientes se ha calculado la exactitud con objeto de comprobar que los datos son correctos, tal y como se ha hecho con el modelo de RL.

$$p = \frac{\sum C_{i=j}}{N} = \frac{233559 + 172926}{233559 + 90940 + 142573 + 172926} = 0.6351$$

$$p = \frac{\sum C_{i=j}}{N} = \frac{27137 + 22168}{27137 + 10928 + 18344 + 22168} = 0.6275$$



Se ha obtenido unos valores similares a los anotados en la Tabla 12 y, por lo tanto, se ha dado paso a evaluar el modelo a través de, nuevamente, las métricas de precisión, exhaustividad, valor F1 y coeficiente Kappa de Cohen. Los datos que se va a utilizar son los extraídos de la matriz de confusión relacionada con los datos de testeo.

### 6.2.2.3. Precisión

En primer lugar se ha calculado la precisión del modelo a la hora de predecir las dos características.

$$precisión_0 = \frac{TN}{TN + FN} = \frac{27137}{27137 + 18344} = 0.5967$$

$$precisión_1 = \frac{TP}{TP + FP} = \frac{22168}{22168 + 10928} = 0.6698$$

A diferencia del modelo RL, en este caso se puede apreciar que la identificación del evento "Derecha" ha sido acertada un mayor número de veces que "izquierda", no logrando llegar ésta última al 60% de aciertos.

### 6.2.2.4. Exhaustividad

Por otro lado, aplicando la fórmula para calcular la exhaustividad o "recall", se ha logrado unos resultados también dispares.

$$exhaustividad_0 = \frac{TN}{TN + FP} = \frac{27137}{27137 + 10928} = 0.7129$$

$$exhaustividad_1 = \frac{TP}{TP + FN} = \frac{22168}{22168 + 18344} = 0.5472$$

En este caso, se ha detectado que el modelo construido por XGboost se deja por predecir un mayor número de veces la categoría "Derecha", equivocándose en un 45% de las veces aproximadamente. Por el contrario, "Izquierda" ha sido clasificado correctamente un 71% de las veces.

### 6.2.2.5. Valor F1

Tal y como se ha podido ver, el modelo tiene una mayor precisión en la categoría positiva pero, sin embargo, una exhaustividad muy inferior a la negativa. Esto se podría interpretar como que, pese a que a la hora de hacer predicciones se deja un mayor número de "Derechas" por clasificar correctamente, las veces que el modelo identifica un evento es de este tipo suele acertar más frecuentemente.

Para ver la valoración del modelo XGboost buscando una cohesión entre precisión y exhaustividad, nuevamente se ha calculado el valor F1.

## Resultados

$$F1_0 = 2 \cdot \frac{\text{precisión}_0 \cdot \text{exhaustividad}_0}{\text{precisión}_0 + \text{exhaustividad}_0} = 2 \cdot \frac{0.5967 \cdot 0.7129}{0.5967 + 0.7129} = 0.6496$$

$$F1_1 = 2 \cdot \frac{\text{precisión}_1 \cdot \text{exhaustividad}_1}{\text{precisión}_1 + \text{exhaustividad}_1} = 2 \cdot \frac{0.6698 \cdot 0.5472}{0.6698 + 0.5472} = 0.6023$$

Los resultados muestran que el modelo funciona casi un 5% mejor al predecir la categoría negativa. A su vez, cabe destacar que en los dos casos el clasificador posee una peor valoración que la dada en el construido por Regresión Logística.

### 6.2.2.6. Coeficiente Kappa de Cohen

Con objeto de comprobar la posibilidad de que el modelo realice clasificaciones de forma aleatoria, al igual que con el modelo de RL, se ha calculado el Coeficiente Kappa de Cohen sobre los resultados del *database* de *Test*.

Para ello, en primer lugar se ha extraído  $p_0$ , indicando que con un valor de 0.4975 el modelo clasificaría de forma completamente aleatoria.

$$p_0 = \frac{\sum C_{i,:} \cdot C_{:,j}}{N^2}$$

$$p_0 = \frac{(27137 + 18344) \cdot (27137 + 10928) + (10928 + 22168) \cdot (18344 + 22168)}{78577^2} = 0.4975$$

Finalmente se ha calculado el coeficiente  $k$ , que en este caso es del 0.2586, siendo una estimación del grado de acuerdo baja y estando cerca de entrar dentro del rango de valores considerados como insignificantes.

$$k = \frac{p - p_0}{1 - p_0} = \frac{0,6275 - 0,4975}{1 - 0,4975} = 0.2586$$

### 6.2.3. Comparación entre los clasificadores

Por medio de este proyecto se ha llevado a cabo, a partir de los datos tomados por un dispositivo MUSE 2, la construcción de varios clasificadores con objeto de predecir unos eventos de Imaginación Motora determinados.

A raíz de los resultados obtenidos al aplicar el modelo sobre el dataset de testeo (*Tabla 19*), se puede sacar la conclusión de que el modelo generado por medio del algoritmo de Regresión Logística tiene una capacidad predictiva superior al de XGboost.

La decisión se ha tomado al haber obtenido unos valores F1 y de la exactitud más elevados. A su vez, los resultados en cuanto a la precisión y la exhaustividad, también han sido superiores en el caso del modelo RL, siendo más parecidos los porcentajes entre la categoría positiva y la negativa, mostrando así un funcionamiento semejante.

Por otro lado, también se puede concluir que el Coeficiente Kappa de Cohen es más elevado en el caso del modelo de la Regresión Logística, manifestando numéricamente que hay una menor probabilidad de realizar clasificaciones de forma aleatoria.

Resultados				
	Modelo de Regresión Logística		Modelo de XGboost	
	Etiqueta 0	Etiqueta 1	Etiqueta 0	Etiqueta 1
<b>Precisión</b>	0.6876	0.6802	0.5967	0.6698
<b>Exhaustividad</b>	0.6354	0.7288	0.7129	0.5472
<b>Valor F1</b>	0.6604	0.703	0.6496	0.6023
	Etiquetas 0 y 1		Etiquetas 0 y 1	
<b>Exactitud</b>	0.6845		0.6275	
<b>Kappa de Cohen</b>	0.365		0.2586	

Etiqueta 0: Imaginación Motora de movimiento de mano hacia la izquierda

Etiqueta 1: Imaginación Motora de movimiento de mano hacia la derecha

**Tabla 19: Resultados finales de cada modelo sobre los datos de testeo**

## 6.3. DISCUSIÓN

### 6.3.1. Valoración del modelo

Tal y como ya se ha demostrado, el clasificador diseñado a partir del algoritmo de Regresión Logística ha sido el más efectivo. Éste, ha conseguido predecir correctamente un número ligeramente menor al 70% de los eventos totales, presentando a su vez una matriz de confusión bastante balanceada.

Es por ello que, de acuerdo a los objetivos planteados de cara a la realización del proyecto, se ha demostrado con éxito la capacidad de poder detectar patrones a partir de señales procesadas *offline* y registradas por medio de un dispositivo EEG del *grado de consumo*.

Sin embargo, tal y como se ha podido demostrar, la probabilidad de que el modelo haya acertado cada uno de los eventos de forma no aleatoria es del 36.5%, asignando una estimación del grado de acuerdo baja (cerca de ser valorada como moderada).

El resultado, por lo tanto, es mejorable, siendo en este punto insuficiente de cara al posible diseño e implementación de una BCI.

### 6.3.2. Hipótesis de mejora

En este punto, se podrían proponer varias hipótesis sobre las fallas del modelo y como poder mejorarlo para lograr clasificaciones más eficaces. A continuación se van a formular estas vías de mejora, dividiéndolas en tres categorías, las que se podría llevar a cabo de forma previa al procesamiento de las señales EEG, durante éste o de forma posterior.

#### 6.3.2.1. Hipótesis para la mejora previa al procesado

En primer lugar, hay que destacar que no ha habido ningún estudio previo capaz de demostrar que los canales utilizados por el dispositivo MUSE 2 guarden correlación con la Imaginación Motora, al no estar situados sobre el área sensorial y motora del cerebro. En consecuencia, las variaciones frecuenciales producidas dentro de la banda beta no se aprecian del mismo modo, captando, posiblemente, información menos útil. A través de un dispositivo EEG que lleve implementados los canales de registro C3 y C4 [66], como, por ejemplo, el diseñado por la compañía NeuroSky, es muy posible que se logre unos resultados mucho mejores al guardar una mayor correlación.

Por otro lado, pese a que el diseño de los ensayos se ha intentado realizar de la forma más correcta posible, por motivos de falta de tiempo, no ha sido posible la automatización de las grabaciones, requiriendo de una segunda persona que oriente al sujeto. De este modo, existe una dependencia de quien controla la interface, pudiendo fallar o distraer al portador del dispositivo de forma involuntaria. Si el ensayo funcionara automáticamente, estos errores no podrían producirse, pasando toda la responsabilidad al sujeto. Por otro lado, el tiempo de registro de cada uno de los eventos sería exacto y uniforme en comparación con el resto, facilitando la normalización de los ensayos y, posiblemente, mejorando la eficacia de técnicas como la de convolución en, por ejemplo, redes neuronales.

Otro tema a tener en cuenta es el número de ensayos. Seguramente 18 sesiones sean una cantidad muy pequeña para poder construir un modelo fiable. La toma de un mayor número de datos con objeto de aumentar el *database* de entrenamiento, podría derivar en dos escenarios:

- Aumentar la capacidad predictiva del modelo al incluir un mayor número de casos posibles.
- Reducirla debido a un sobrentrenamiento, también conocido en inglés por el término de “*Overfitting*”.

Finalmente, también se deberían evaluar parámetros del ensayo como los tiempos de registro y etiquetado de cada evento, o la duración total de la prueba, ya que igual 4

minutos es un tiempo excesivo para mantener al sujeto concentrado en la tarea. Hay que tener en cuenta que se debe dar comodidad a éste, ya que si se descentra, las señales pueden alterarse en gran medida, haciendo el ensayo inservible.

#### ***6.3.2.2. Hipótesis para la mejora en el procesado***

Durante la etapa del procesado de la señal en bruto, un punto importante por el cual el resultado puede oscilar en gran medida es la aplicación de la técnica ICA. Un reajuste de los parámetros que regulan el algoritmo podría evitar que se eliminara información trascendente. Por otro lado, cuanto mayor sea el número de canales de registro, la señal EEG se podrá dividir en más ICs, reduciendo también de este modo las posibilidades de filtrar datos importantes para la búsqueda de patrones.

A su vez, hay que tener en cuenta que la etapa de extracción de características tiene la mayor importancia a la hora de entrenar un modelo. Es por ello que habría que evaluar otras técnicas vistas de forma previa en el *apartado 4.2.2.3.2*, ya sean del dominio del tiempo, de la frecuencia o ambos. Entre otras, revisando estudios anteriores destacaría la *Transformada de Wavelet* o el cálculo del *Espectro de Densidad de Potencia*, al ser, de entre todos, los métodos más utilizados para la detección de variaciones en la actividad sensoriomotora.

#### ***6.3.2.3. Hipótesis para la mejora tras el procesado***

Tras la etapa de procesado, tal y como se ha visto, se llega al punto de entrenar el clasificador a partir de los datos del *dataset* de entrenamiento, comprobando después su eficacia con el de testeo.

Dentro del presente Trabajo de Fin de Grado, el tipo de algoritmo empleado para la construcción de los modelos ha sido *lineal* en los dos casos. Por lo tanto, una alternativa a tener en cuenta, a la hora de probar a mejorar los resultados, sería la utilización de técnicas *no lineales*, también conocidas como redes neuronales (simples o de aprendizaje profundo). Entre otras, en varios estudios se ha destacado la buena labor de las redes convolucionales (CNN), pudiéndose encontrar en Internet varias ya predefinidas. Para su aplicación cabe destacar que, probablemente, habría que cambiar el tipo de ensayo con el cual grabar las señales a procesar.

Por otro lado, es posible que haya otro tipo de técnicas lineales capaces de lograr un mejor resultado. En estos casos, sin embargo, el tiempo necesario para entrenar el clasificador podría ser mucho mayor, llegando a superar en algún caso las dos horas.

## 7. CONCLUSIONES

### 7.1. CONCLUSIONES GENERALES

A lo largo del presente Trabajo de Fin de Grado, se ha realizado una investigación acerca del campo de las BCIs basadas en señales EEG. Más concretamente, se ha realizado el ajuste de varios modelos predictivos con objeto de formar un sistema BCI que sea capaz de controlar un sistema informático. Los resultados obtenidos han demostrado la viabilidad de utilizar un dispositivo comercial MUSE 2 para la adquisición de señales EEG a través de sus cuatro canales de registro dentro de un BCI.

Es conveniente destacar que para llevar a cabo tal tarea, se ha logrado construir una base de datos que contenga la información de las señales EEG correctamente etiquetadas en función del evento de Imaginación Motora que se esté registrando.

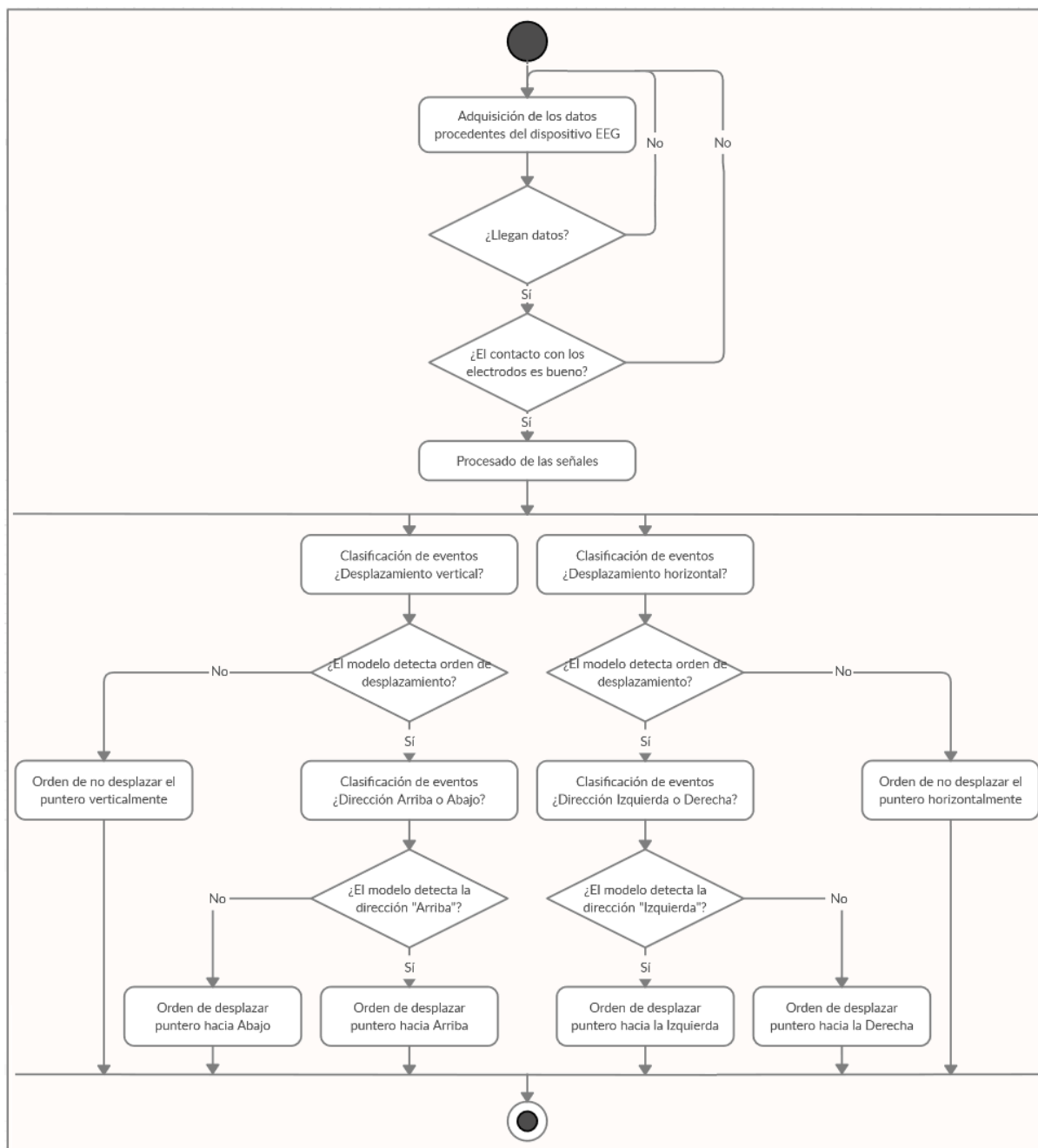
### 7.2. LÍNEA FUTURA

Una vez se consiga obtener un modelo que, a la hora de clasificar los datos de *Test*, presente un Coeficiente Kappa de Cohen, como mínimo, superior al 0.6, se debería tratar de replicar el mismo procedimiento para predecir las siguientes parejas de eventos:

- “Desplazamiento de mano horizontalmente” / “No hay desplazamiento”
- “Desplazamiento de mano verticalmente” / “No hay desplazamiento”
- Desplazamiento vertical de mano “hacia arriba” / “hacia abajo”

Una vez logrados unos resultados óptimos, se podría pasar a la siguiente etapa, la construcción de la BCI. Para ello, se debería implementar el código ya realizado para el procesamiento *offline* de la señal para que funcione a tiempo real. Un software apropiado para esta tarea, por ejemplo, sería LabVIEW, a través del cual se podría diseñar un programa que permita construir un prototipo para probar el funcionamiento de la BCI planteada.

Ésta podría funcionar según la estructura que se ha planteado en el diagrama UML de la Ilustración 52, en el cual, tras adquirir y procesar los datos, pasarían por cuatro modelos diferentes. De este modo, el puntero se podría mover en vertical, horizontal, diagonal (los modelos predicen desplazamiento vertical y horizontal) o quedarse estático.



**Ilustración 52: Diagrama UML de Actividad de la BCI planteada**

## 8. BIBLIOGRAFÍA

[1] X. Gu *et al.*, «EEG-based Brain-Computer Interfaces (BCIs): A Survey of Recent Studies on Signal Sensing Technologies and Computational Intelligence Approaches and their Applications», *arXiv:2001.11337 [cs, eess]*, ene. 2020, Accedido: feb. 26, 2020. [En línea]. Disponible en: <http://arxiv.org/abs/2001.11337>.

[2] Lauren Golembiewski, «Are You Ready for Tech That Connects to Your Brain?», *HBR*, [En línea]. Disponible en: <https://hbr.org/2020/09/are-you-ready-for-tech-that-connects-to-your-brain>.

[3] «Neuralink». <https://neuralink.com/> [Accedido: 15-oct-2020].

[4] G. M. Rojas, C. Alvarez, C. E. Montoya, M. de la Iglesia-Vayá, J. E. Cisternas, y M. Gálvez, «Study of Resting-State Functional Connectivity Networks Using EEG Electrodes Position As Seed», *Front. Neurosci.*, vol. 12, p. 235, abr. 2018, doi: 10.3389/fnins.2018.00235.

[5] Roberto Hornero, Rebeca Corralejo, y Daniel Álvarez, «Brain-Computer Interface (BCI) aplicado al entrenamiento cognitivo y control domótico para prevenir los efectos del envejecimiento», *F. G. CSIC*, [En línea]. Disponible en: [http://www.fgcsic.es/lychnos/es\\_es/articulos/Brain-Computer-Interface-aplicado-al-entrenamiento-cognitivo](http://www.fgcsic.es/lychnos/es_es/articulos/Brain-Computer-Interface-aplicado-al-entrenamiento-cognitivo).

[6] F. Ramos-Argüelles, G. Morales, S. Egozcue, R. M. Pabón, y M. T. Alonso, «Técnicas básicas de electroencefalografía: principios y aplicaciones clínicas», *An. Sist. Sanit. Navar.*, vol. 32, p. 14, 2009.

[7] «International Federation of Clinical Neurophysiology». <https://www.ifcn.info/index.asp> [Accedido: 15-oct-2020].

[8] Universidad Militar Nueva Granada, R. Jiménez Moreno, J. Rodríguez Alemán, y Universidad Militar Nueva Granada, «Control of a Mobile Robot Through Brain Computer Interface», *Inge cuc*, vol. 11, n.º 2, pp. 74-83, jul. 2015, doi: 10.17981/ingecuc.11.2.2015.08.

[9] «Overview of artifact detection - MNE 0.22.dev0 documentation». [https://mne.tools/dev/auto\\_tutorials/preprocessing/plot\\_10\\_preprocessing\\_overview.html](https://mne.tools/dev/auto_tutorials/preprocessing/plot_10_preprocessing_overview.html) [Accedido: 20-sept-2020].



- [10] «Repairing artifacts with ICA - MNE 0.21.dev0 documentation». [https://mne.tools/stable/auto\\_tutorials/preprocessing/plot\\_40\\_artifact\\_correction\\_ica.html](https://mne.tools/stable/auto_tutorials/preprocessing/plot_40_artifact_correction_ica.html) [Accedido: 20-sept-2020].
- [11] S. Pathirana, D. Asirvatham, y G. Johar, «A Critical Evaluation on Low-Cost Consumer-Grade Electroencephalographic Devices», en *2018 2nd International Conference on BioSignal Analysis, Processing and Systems (ICBAPS)*, Kuching, jul. 2018, pp. 160-165, doi: 10.1109/ICBAPS.2018.8527413.
- [12] E. Cutrell y D. Tan, «BCI for passive input in HCI», p. 3.
- [13] International Conference on Artificial Intelligence, *ICAI 2016: proceedings of the 2016 International Conference on Artificial Intelligence: WORLDCOMP'16, July 25-28, 2016, Las Vegas, Nevada, USA*. U.S.A.: CSREA Press, 2016.
- [14] O. E. Krigolson, C. C. Williams, A. Norton, C. D. Hassall, y F. L. Colino, «Choosing MUSE: Validation of a Low-Cost, Portable EEG System for ERP Research», *Front. Neurosci.*, vol. 11, mar. 2017, doi: 10.3389/fnins.2017.00109.
- [15] «Muse™ - medita con facilidad». <https://choosemuse.com/es/> [Accedido: 12-sept-2020].
- [16] M. A. G. Félix-López, A. E. Sánchez-Martínez, y L. M. Alonso-Valerdi, «Actividad Neuronal Prefrontal durante Pruebas Lógico-Matemáticas en Estudiantes que realizan Actividad Física Continua y Aquellos que son Sedentarios», p. 5, 2018.
- [17] «Mind Monitor». <https://mind-monitor.com/> [Accedido: 12-sept-2020].
- [18] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, y T. M. Vaughan, «Brain-computer interfaces for communication and control», *Clinical Neurophysiology*, p. 25, 2002.
- [19] A. Bashashati, M. Fatourehchi, R. K. Ward, y G. E. Birch, «A survey of signal processing algorithms in brain-computer interfaces based on electrical brain signals», *J. Neural Eng.*, vol. 4, n.º 2, pp. R32-R57, jun. 2007, doi: 10.1088/1741-2560/4/2/R03.
- [20] N. Birbaumer *et al.*, «The thought translation device (TTD) for completely paralyzed patients», *IEEE Trans. Rehab. Eng.*, vol. 8, n.º 2, pp. 190-193, jun. 2000, doi: 10.1109/86.847812.
- [21] I. Iturrate, J. M. Antelis, A. Kubler, y J. Minguez, «A Noninvasive Brain-Actuated Wheelchair Based on a P300 Neurophysiological Protocol and Automated Navigation», *IEEE Trans. Robot.*, vol. 25, n.º 3, pp. 614-627, jun. 2009, doi: 10.1109/TRO.2009.2020347.

## Bibliografía

[22] L. A. Farwell y E. Donchin, «Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials», *Electroencephalography and Clinical Neurophysiology*, vol. 70, n.º 6, pp. 510-523, dic. 1988, doi: 10.1016/0013-4694(88)90149-6.

[23] «Donchin et al. - 2000 - The mental prosthesis assessing the speed of a P3.pdf». .

[24] M. Spuler, «A Brain-Computer Interface (BCI) system to use arbitrary Windows applications by directly controlling mouse and keyboard», en *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Milan, ago. 2015, pp. 1087-1090, doi: 10.1109/EMBC.2015.7318554.

[25] «¿Que es LabVIEW? - NI». <https://www.ni.com/es-es/shop/labview.html> [Accedido: 14-sept-2020].

[26] «¿Como funciona el Bluetooth?» <https://comofuncionaque.com/como-funciona-el-bluetooth/> [Accedido: 30-oct-2020].

[27] «Cómo funciona Bluetooth Low Energy: el protocolo estrella de IoT». <https://www.welivesecurity.com/la-es/2020/03/17/como-funciona-bluetooth-low-energy/> [Accedido: 30-oct-2020].

[28] «UDP: ¿qué es el protocolo UDP? - Ionos». <https://www.ionos.es/digitalguide/servidores/know-how/udp-user-datagram-protocol/> [Accedido: 30-oct-2020].

[29] «Introducción al protocolo de comunicación Open Sound Control». <https://mosaic.uoc.edu/2019/04/03/introduccion-al-protocolo-de-comunicacion-open-sound-control/>.

[30] «Introducción al OSC (Open Sound Control)». <https://www.inventable.eu/2010/08/03/introduccion-al-osc-open-sound-control-primer-parte/> [Accedido: 30-oct-2020].

[31] A. N. Belkacem *et al.*, «Real-Time Control of a Video Game Using Eye Movements and Two Temporal EEG Sensors», *Computational Intelligence and Neuroscience*, vol. 2015, pp. 1-10, 2015, doi: 10.1155/2015/653639.

[32] M. Agarwal y R. Sivakumar, «Blink: A Fully Automated Unsupervised Algorithm for Eye-Blink Detection in EEG Signals», en *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, USA, sep. 2019, pp. 1113-1121, doi: 10.1109/ALLERTON.2019.8919795.

- [33] E. G. Gutiérrez, «Introducción al filtrado digital», p. 14.
- [34] B. Medina, J. E. Sierra, y A. B. Ulloa, «Técnicas de extracción de características de señales EEG en la imaginación de movimiento para sistemas BCI», p. 13.
- [35] P. Bashivan, I. Rish, y S. Heisig, «Mental State Recognition via Wearable EEG», p. 10.
- [36] F. Lotte y Cuntai Guan, «Regularizing Common Spatial Patterns to Improve BCI Designs: Unified Theory and New Algorithms», *IEEE Trans. Biomed. Eng.*, vol. 58, n.º 2, pp. 355-362, feb. 2011, doi: 10.1109/TBME.2010.2082539.
- [37] R. H. Gatti, L. G. Escher, y L. C. Carrere, «Interfaz Cerebro Computadora basada en OpenVIBE y el Dispositivo EPOC: Resultados Preliminares», p. 6.
- [38] «Convolución y Correlación». <http://www6.uniovi.es/vision/intro/node30.html> [Accedido: 30-oct-2020].
- [39] Joaquín Amat Rodrigo, «Regresión Logística y Múltiple», ago. 2016. [https://www.cienciadedatos.net/documentos/27\\_regresion\\_logistica\\_simple\\_y\\_multiple](https://www.cienciadedatos.net/documentos/27_regresion_logistica_simple_y_multiple).
- [40] María Elvira Ferre Jaén, «FEIR 45: Regresión logística». [https://gauss.inf.um.es/feir/45/#2\\_regresi%C3%B3n\\_log%C3%ADstica\\_binaria](https://gauss.inf.um.es/feir/45/#2_regresi%C3%B3n_log%C3%ADstica_binaria). [Accedido: 2-nov-2020].
- [41] Juan Bosco Mendoza Vega, «XGBoost en R», dic. 08, 2019. <https://medium.com/@jboscomendoza/xgboost-en-r-398e7c84998e>. [Accedido: 2-nov-2020].
- [42] J. J. Espinosa Zúñiga, «Aplicación de algoritmos Random Forest y XGBoost en una base de solicitudes de tarjetas de crédito», *iit*, vol. 21, n.º 3, pp. 1-16, jul. 2020, doi: 10.22201/fi.25940732e.2020.21.3.022.
- [43] «Como funciona Gradient Boosting», abr. 10, 2019. <https://spainml.com/blog/como-funciona-gradient-boosting/>. [Accedido: 2-nov-2020].
- [44] «Introduction to Boosted Trees». <https://xgboost.readthedocs.io/en/latest/tutorials/model.html>. [Accedido: 2-nov-2020].
- [45] T. Yu, Y. Li, J. Long, y Z. Gu, «Surfing the internet with a BCI mouse», *J. Neural Eng.*, vol. 9, n.º 3, p. 036012, jun. 2012, doi: 10.1088/1741-2560/9/3/036012.
- [46] S. Sree Shankar y R. Rai, «Human factors study on the usage of BCI headset for 3D CAD modeling», *Computer-Aided Design*, vol. 54, pp. 51-55, sep. 2014, doi: 10.1016/j.cad.2014.01.006.

## Bibliografía

- [47] T. McMahan, I. Parberry, y T. D. Parsons, «Modality specific assessment of video game player's experience using the Emotiv», *Entertainment Computing*, vol. 7, pp. 1-6, mar. 2015, doi: 10.1016/j.entcom.2015.03.001.
- [48] J. J. Bird, «Mental Emotional Sentiment Classification with an EEG- based Brain-Machine Interface», *DISP* ', p. 7.
- [49] J. Teo y J. T. Chia, «Deep Neural Classifiers For Eeg-Based Emotion Recognition In Immersive Environments», en *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, Shah Alam, jul. 2018, pp. 1-6, doi: 10.1109/ICSCEE.2018.8538382.
- [50] J. L. Sirvent, J. M. Azorín, E. Iáñez, A. Úbeda, y E. Fernández, «Interfaz Cerebral no Invasiva Basada en Potenciales Evocados para el Control de un Brazo Robot», *Rev. iberoam. autom. inform. ind.*, vol. 8, n.º 2, pp. 103-111, abr. 2011, doi: 10.4995/RIAI.2011.02.13.
- [51] P. Peining, G. Tan, y A. A. P. Wai, «Evaluation of Consumer-Grade EEG Headsets for BCI Drone Control», p. 6.
- [52] A. Campos y M. A. González, «Versión Española del Cuestionario-Revisado de Imagen del Movimiento (MIQ-R): Validación y Propiedades Psicométricas», p. 13.
- [53] «Precision, Recall, F1, Accuracy en clasificación - IArtificial.net». <https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/>. [Accedido: 5-nov-2020].
- [54] «Kappa de Cohen | SAMIUC». <http://www.samiuc.es/estadisticas-variables-binarias/medidas-de-concordancia/kappa-de-cohen/>. [Accedido: 3-nov-2020].
- [55] «Wireshark · Go Deep.». <https://www.wireshark.org/>.
- [56] «Biblioteca NI OSC para LabVIEW». <https://sine.ni.com/nips/cds/view/p/lang/es/nid/211428>. [Accedido: 16-sep-2020].
- [57] «Welcome to Python.org». <https://www.python.org/>. [Accedido: 20-sep-2020].
- [58] «Anaconda | Individual Edition». <https://www.anaconda.com/products/individual>. [Accedido: 20-sep-2020].
- [59] «pandas - Python Data Analysis Library». <https://pandas.pydata.org/>. [Accedido: 20-sep-2020].
- [60] «Google Colab». <https://colab.research.google.com>. [Accedido: 20-sep-2020].

- [61] «MNE — MNE 0.21.0 documentation». <https://mne.tools/stable/index.html>. [Accedido: 20-sept-2020].
- [62] «Matplotlib: Python plotting — Matplotlib 3.3.2 documentation». <https://matplotlib.org/>. [Accedido: 20-sept-2020].
- [63] «SciPy.org — SciPy.org». <https://www.scipy.org>. [Accedido: 20-sept-2020].
- [64] «NumPy». <https://numpy.org/>. [Accedido: 20-sept-2020].
- [65] «scikit-learn: machine learning in Python — scikit-learn 0.23.2 documentation». <https://scikit-learn.org/stable/>. [Accedido: 20-sept-2020].
- [66] M. H., A. AbuBaker, A. Turani, y A. M., «EEG Mouse: A Machine Learning-Based Brain Computer Interface», *IJACSA*, vol. 5, n.º 4, 2014, doi: 10.14569/IJACSA.2014.050428.



## **Relación de documentos**

(X) Memoria	89	páginas
(_) Anexos	46	páginas

La Almunia, a 25 de noviembre de 2020

Firmado: Iñigo Garaboa Cotelo